①

AFIT/GOR/ENS/93M-13

THE AIRLIFT CAPABILITIES ESTIMATION PROTOTYPE: A CASE STUDY IN MODEL VALIDATION

THESIS

Randy McCanne, Captain, USAF

AFIT/GOR/ENS/93M-13

Approved for public release; distribution unlimited

AFIT/GOR/ENS/93M-13

THE AIRLIFT CAPABILITIES ESTIMATION PROTOTYPE: A CASE STUDY IN MODEL VALIDATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Randy McCanne, B.S.
Captain, USAF

March 1993

THESIS APPROVAL

STUDENT: Capt Randy McCanne CLASS: GOR - 93M

THESIS TITLE: THE AIRLIFT CAPABILITIES ESTIMATION PROTOTYPE: A CASE STUDY IN MODEL VALIDATION

DEFENSE DATE: 26 February 1993

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Dr. J. W. Chrissis/ENS	
Reader	Lt Col J. T. Moore/ENS	James T. Moore

Preface

This study was initiated with the goal of developing a more mature framework for building and validating mathematical models. Since modeling is often the most cost effective way to investigate the impact of decisions, it is likely to become even more important in this decade of "do more with less". Most models in use today were developed "the old fashion way" where the analyst assigned to a problem develops a model formulation which is passed to other analysts who improve upon it. Thus, the model evolves through time into a useful decision-making tool (maybe). Unfortunately, the usual result of this process of model evolution is a complicated patchwork model that seems to provide good answers but no one knows how or why.

Since the goal of modeling is insight – not numbers – it is imperative that the models developed for the Air Force be useful, understandable, and maintainable. This requires a controlled process with more emphasis on requirements, documentation, and configuration management. Thus, the life cycle approach is born. Since the life cycle of a model is a *process*, the next step on the "maturity ladder" might be called *process control*. I leave this topic for follow on research.

I am indebted to my faculty advisor, Dr. Chrissis, for his patience and expertise, and to my reader, Lt Col Moore, for his very helpful constructive criticism. I also wish to thank Lt Col Litko and Mr. Alan Whisman of the Force Structure Analysis Division at Air Mobility Command for their outstanding help in working with the ACEP model and obtaining the Desert Shield data. Despite my initial skepticism, Lt Col Litko's predictions on the improvements needed in the model turned out to be amazingly accurate. Finally, I must thank my wife, Susan, for her patience and understanding while I was buried in my thesis work. A better wife no man ever had.

Randy McCanne

Table of Contents

List of Figures

List of Tables

AFIT/GOR/ENS/93M-13

Abstract

This study investigates the application of a life cycle approach to the validation of operational models. The classic "waterfall" life cycle from software engineering is adapted for use on mathematical models by defining four stages of model development. Each stage is discussed in detail and examples of the output from each stage are presented. In addition, techniques are investigated for applying the proposed life cycle to existing models through the recovery of life cycle stages.

The methodology is applied to a linear programming model developed for planning airlift operations to demonstrate the power of the life cycle approach to validation. The results of applying each stage of the life cycle to the model are presented. As a final test, the model is used to predict the airlift capability and resource requirements for the Operation Desert Shield airlift. A comparison is made between the predictions of the model and data from the actual operation. The validated model is shown to be a better representation of the airlift planning problem. Finally, specific recommendations are made for operational use of the airlift planning model and on areas where further research is needed on both the model and the life cycle validation approach.

THE AIRLIFT CAPABILITIES ESTIMATION PROTOTYPE: A CASE STUDY IN MODEL VALIDATION

I. *Introduction*

Problem Statement

The effective projection of combat power over great distances is now, more than ever, an important task of the United States armed forces. The Air Force plays a crucial role in achieving this goal through the employment of airlift forces. Yet with a shrinking budget and force size, improvements in force projection capability must come from more effective use of aircraft and crews. One way to achieve this is through the efficient scheduling and routing of aircraft during an airlift operation. Air Force planners have sophisticated tools available to help develop airlift plans for specific scenarios, but the massive airlift operation of Desert Shield exposed weaknesses in these tools. The tools were unable to keep pace with the fast-changing requirements and priorities of the early stages of the operation. The purpose of this thesis is to examine the Airlift Capabilities Estimation Prototype (ACEP), a model proposed as a solution to this weakness, and investigate how well it meets the airlift planning challenge.

Background

Importance of Strategic Mobility. Future conflicts are likely to be short, violent, and a long way from U.S. shores. The Persian Gulf deployment was the prototype. The role of transportation, while an important aspect of any military action, was made starkly visible this time. "Anytime we have to take an action, we will have to move a force very, very quickly. From a strategy standpoint, I see transportation being of increased importance" says General H.T. Johnson, commander in chief of U.S. Transportation Command [Powell, 1991:52]. U.S. strategic mobility forces moved some 35,000 troops and 1 billion pounds of cargo in the first two weeks of the Persian Gulf deployment, helping deter Iraq from attacking Saudi Arabia. "The United States projected forces, equipment, and sustainment farther, faster and in greater quantities than ever before." [Conduct of the Persian Gulf War, 1992:E-6].

Recent world events have increased the importance of strategic mobility. The unparalleled metamorphosis of our old nemesis, the former Union of Soviet Socialist Republics, has precipitated incredible changes in the entire world and in our requirements for military capability. Some of these changes include [Bossert, 1990:3 - 4]:

- Increased importance of conventional forces. The end of the cold war has brought greater willingness to reduce nuclear weapon stockpiles. A decrease in the deterrence value of nuclear weapons makes a strong conventional capability more important.
- Proliferation of third world "hot spots". The number and intensity of ongoing or potential conflicts throughout the world will continue to grow as regions adjust to the vacuum created by the decline of Soviet influence.
- Reduction of overseas bases. The economic and budgetary challenges of the U.S., the increasing reluctance of Americans to fund the defense of Europe and Japan, the widespread perception of a decrease in the threat posed by the former Soviet Union, and the increased reluctance of our allies to renew basing rights will almost certainly result in a decrease in the number and size of overseas bases.

The Strategic Mobility Triad. All of these changes point to the requirement for a military force that is small, flexible, and more mobile than ever before. The U.S. depends upon a triad of mobility capabilities to project forces to a threatened area. Each leg of the triad – airlift, sealift and prepositioning – has unique strengths and weaknesses that, when properly balanced, can provide the projection capability necessary for each stage of a conflict (Figure 1). Airlift is the fastest and most flexible leg of the triad, but it is very limited in capacity. In Desert Shield, for example, only about 5 percent of the total cargo was delivered by air [Conduct of Persian Gulf War, 1992:E - 9]. Prepositioning is an attractive option when the location of the next conflict can be accurately forecast but is of limited value otherwise [Miller, 1988:373]. Sealift can move huge quantities of materials, but it is slow. Experts estimate that sealift to some regions of the world may take as long as 30 to 40 days which can more than offset the advantage in capacity

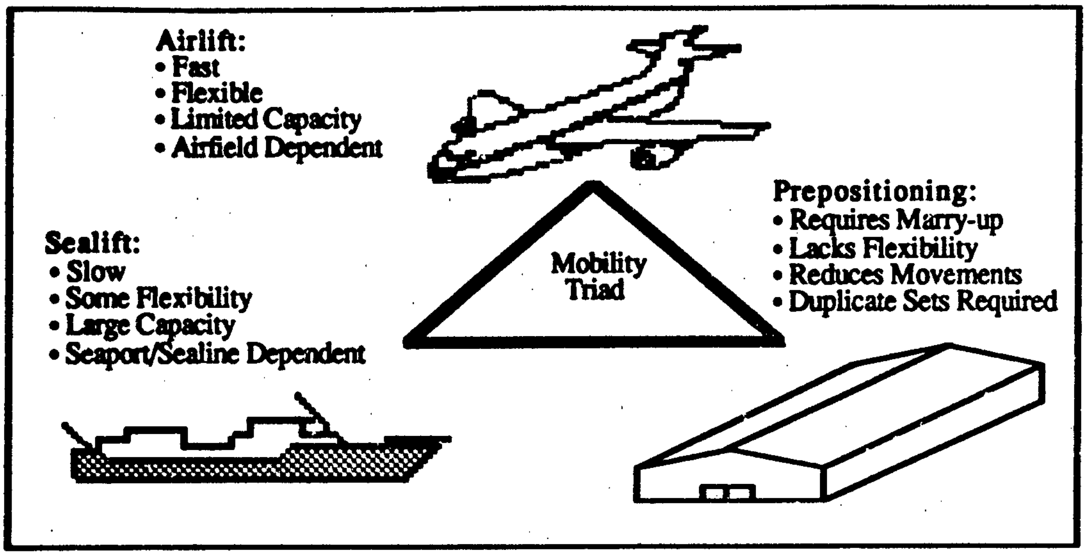


Figure 1. Balanced Mobility (Adapted from Miller, 1988: 366)

[King, 1989:1]. As an example, the first American cargo ship that arrived in Israel during the 1973 Mid-East war delivered more tonnage than was delivered by airlift in the previous 30 days. However, the war had been over for 20 days [Comptroller General, 1975].

The Airlift Planning System. Airlift provides the critical projection capability in the first few days of a conflict; usually before a clear plan of action has been determined. Consequently, an airlift planning system must be flexible enough to handle daily or even hourly changes in movement requirements and priorities. Air Force regulation 28-3 defines the Joint Operational Planning and Execution System (JOPES) as the single, integrated system for joint planning within the Department of Defense (Figure 2). JOPES provides an automated system for use in both deliberate and execution planning.

Deliberate planning is the process of developing Concept Plans (CONPLANS) and Operation Plans (OPLANS) to support national security policy. OPLANS and their accompanying Time-Phased Force and Deployment Data (TPFDD or "tip-fid") are developed in anticipation of a future airlift operation. The OPLAN describes the aircraft, airfield and other resources assigned to support an airlift operation. The TPFDD contains deployment data, including on-load and off-load airfields, and the type and amount of cargo and personnel to be deployed. Concept Plans provide the flexibility and rapid reaction needed during contingency situations. CONPLANS do not have corresponding TPFDDs, but contain a summary of the

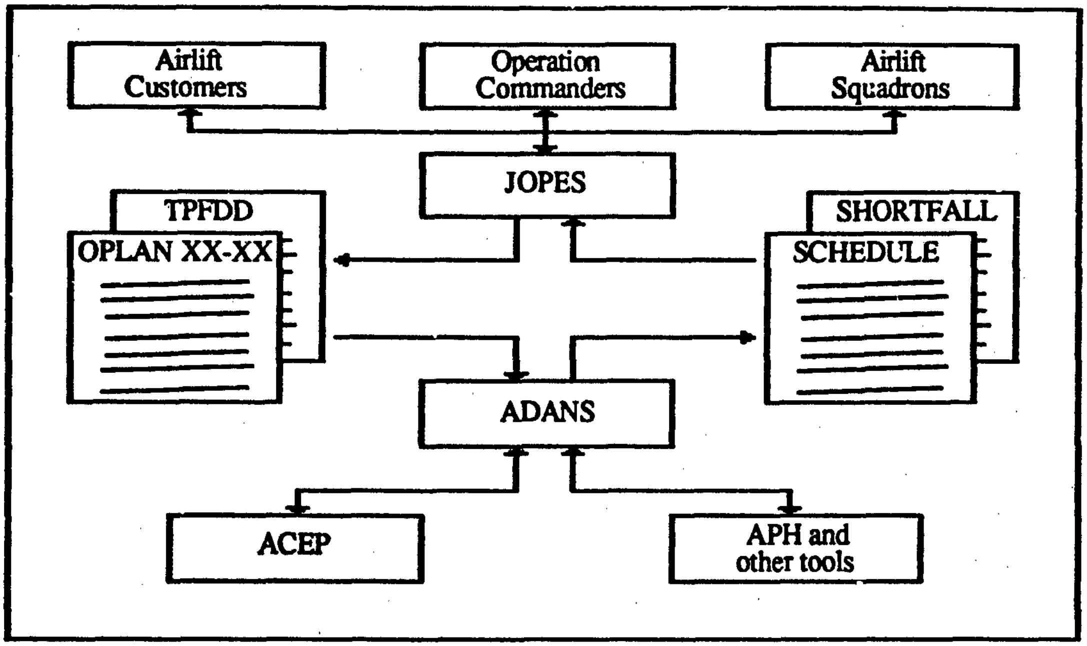


Figure 2. Airlift Planning System

most likely mobility and logistic assets needed to support the execution of a plan. The information available at the time of deliberate planning is necessarily rough and incomplete. Consequently, the goal of deliberate planning is not to produce a detailed schedule of operations but only to provide a starting point for execution planning [Rappoport et al., 1992:75; Rappoport et al., 1991:64].

Execution planning is conducted in response to an actual airlift requirement – such as an exercise, contingency, or humanitarian relief effort. A very powerful tool is required to match the individual airlift requirements with aircraft and crews, and then to schedule individual missions to perform the airlift. Normally, enough information is available during execution planning to schedule several days in advance, but the schedule must be continuously adjusted so as to account for changing requirements and resources [Rappoport et al., 1992:75].

Lessons from Desert Shield. One of the most important lessons learned from the Persian Gulf conflict was the importance of sound planning in the employment of strategic lift assets (airlift and sealift) [Conduct of the Persian Gulf War, 1992:xxiv]. While advance planning played an important role in the overall success of the U.S. response, airlift planners were largely unable to adapt existing plans to the rapidly changing situation in the time required. According to

Major Bruce Babb, a member of the Military Airlift Command (MAC) Crisis Action Team during Desert Shield, the planning systems and methods in use in August 1990 were not flexible enough to handle the airlift requirements of operation Desert Shield. "The priorities kept changing – sometimes six times a day – while the deployment flow was going on." [Babb, 1990:1]. JOPES was unable to cope with the constant changes in airlift requirements. Just sorting out the requirements and priorities of the massive airlift for input to the system could have delayed the first airlift mission by weeks or even months. Since even a week's delay was unacceptable, the MAC planners were forced to abandon use of much of JOPES and manually control the flow of aircraft [Babb, 1990:1 - 3].

In addition, deployment data had not been reviewed to determine transportation feasibility. A transportation feasibility study determines the assets needed to move the personnel and equipment of a specific military unit. Rapid response units were the only ones for which current transportation feasibility data was available. As a result, transportation planners were forced to improvise, and airlift requirements exceeded capability by as much as 7,000 tons per day [Conduct of the Persian Gulf War, 1992:E - 3; Babb, 1990:2].

The Development of ADANS. The weaknesses in JOPES were identified long before Desert Shield highlighted them. Development of a system to augment JOPES, called the Airlift Deployment Analysis System (ADANS), was started in 1987. ADANS is an interactive database system with an array of tools which can be used to perform both deliberate and execution planning on a large scale. The centerpiece of ADANS' automated scheduling tools is a dynamic programming-based algorithm called the airlift-planning heuristic (APH) [Hilliard et al., 1992:135]. The objective of APH is to develop an airlift schedule that maximizes the on-time delivery of cargo and passengers . The APH is a very powerful tool and is capable of scheduling 10,000 missions in under two hours [Rappoport et al., 1992: 86].

ADANS was under development when Desert Shield started, and the program was accelerated to help cope with the inadequacies of the JOPES system. Because of its rushed implementation, no performance comparison of ADANS to JOPES was made, but there is little doubt that it contributed significantly to the success of the nearly $4 billion airlift operation [Hilliard et al., 1992:140] However, development of the deliberate planning tools within

ADANS was delayed in favor of the development of the execution planning tools, such as the APH, urgently needed for Desert Shield. Consequently, many tools are still under development and the ADANS system as a whole is not expected to reach initial operational capability until March 1993 [Mitchell, 1992].

The Airlift Capabilities Estimation Prototype (ACEP) model was developed by Busch and Hilliard of the Operations Research Group, Oak Ridge National Laboratory, as part of ADANS to support deliberate planning. It is a linear programming-based tool designed to provide quick estimates of the resources needed to support an airlift operation [Busch and Hilliard, 1992:1]. ACEP is designed to be fast, flexible, and work with the rough information that is available during deliberate planning or the early stages of execution planning. The model has the potential to address the problems encountered in the early days of Desert Shield, but a formal analysis is required to verify that the model is accurate enough to be of practical use.

Research Objectives

The ACEP model was developed in response to a demonstrated need for a tocl to provide quick estimates of the resources required to support a planned airlift operation. However, this model has not been validated to ensure that it adequately represents the airlift planning problem. The purpose of this effort is to validate the ACEP model and to improve the model, if possible, based upon the validation findings. Specifically, two main problems are addressed by this research:

(1) How to independently evaluate the validity of an existing model.

(2) How to improve the ACEP model to better meet the needs of Air Mobility Command in solving their airlift planning problems.

The focus of the first part of the research is on the validation of mathematical models. While the validation methodology developed is demonstrated using a linear programming-based model, the general approach is applicable to other forms of models as well (e.g. nonlinear models, simulation models, etc.). The goal of this research is to develop and demonstrate a structured, methodical approach to the validation of existing models.

The focus of the second part of the research is to address questions presented by R.D. Specht in his discussion on model testing in "The Nature of Models" [Specht, 1968:220]:

(1) Can the ACEP model describe correctly and clearly the known facts and situations?

(2) When the principal parameters involved are varied, do the results remain consistent and plausible?

(3) Can the ACEP model handle special cases in which there is some indication as to what the outcomes should be?

(4) Can it assign causcs to known effects?

Overview of Subsequent Chapters

Chapter II contains a summary of published literature on vehicle routing and scheduling problems and a review of techniques commonly used in operations research to validate models. In addition, a review of validation techniques in software engineering is presented and the applicability of these techniques to mathematical model validation is discussed.

Chapter III presents a model development life cycle and the results of applying the first two stages of the life cycle to the ACEP model. A recovery process is used to reconstruct the life cycle stages that led to the existing model design. Some significant improvements are made to the model as a result of the life cycle recovery process.

In Chapter IV, the operational (executable) version of ACEP is presented. The techniques of constraint validation are used to ensure the model accurately represents the conceptual ACEP model design. Again, improvements are made to the model design as a result of the constraint validation process.

Chapter V presents the results of the specific post-development validation technique employed. A retrospective (or predictive) test is conducted using data from Operation Desert Shield. The test proved to be highly useful in providing insights into the model's validity for use in planning and executing sustained airlift operations.

Chapter VI concludes the research and provides recommendations for further research.

II. *Literature Review*

The airlift planning problem can be thought of as a vehicle routing problem (VRP) with both time and capacity constraints [Rappoport et al., 1992:74]. In the first part of this chapter, an overview of the VRP is presented including classification schemes and solution approaches. The second part of the chapter reviews verification and validation techniques used on both mathematical models and computer software. Useful parallels are drawn to aid in the validation of the ACEP model.

Vehicle Routing and Scheduling Problems

> The costs associated with operating vehicles and crews for delivery purposes form an important component of total distribution costs. Consequently small percentage savings in these expenses could result in substantial total savings over a number of years . . . The use of analytic routing and scheduling models and techniques can be instrumental in realizing the savings . . . [Bodin et al., 1983:70]

In general, a VRP can be defined as: A set of customers, each with a known location and a known requirement for some commodity, is to be supplied from a set of depots by a set of delivery vehicles. Instantiations of the VRP vary widely, and may differ in the number of vehicles, customers and depots. Most practical VRPs also contain some time and capacity constraints. In the case of a single vehicle with unlimited capacity, the problem reduces to the well-known traveling salesman problem. The objective of modeling VRPs is to develop an "optimal" route or schedule for each vehicle. Bodin, Golden, Assad and Ball give an overview of vehicle routing problems in a special 1983 edition of *Computers and Operations Research* [Bodin et al., 1983].

Classification of VRPs. Several classification schemes for VRPs have been proposed. Bodin et al. provide a classification of VRPs into three groups: (1) pure routing, (2) pure scheduling, and (3) a combination of both routing and scheduling. These groups are then subdivided into a more detailed classification.

More recently, Desrochers, Lenstra and Savelsbergh developed a classification scheme based upon the various constraints added to the basic problem [Desrochers et al., 1990]. Their scheme classifies VRPs based upon the characteristics of the customers, the vehicles, the service strategies employed, and the objective of the model.

In many cases, customers can only be serviced during specified time windows. Solomon and Desrosiers classify the different types of VRPs with time windows by the underlying mathematical model that most closely matches the problem [Solomon and Desrosiers, 1988]. They define eight classes of problems. The difficulty with this approach is that one must be able to determine the underlying mathematical model from the problem description. However, once the underlying model is determined, it can lead directly to a set of published solution algorithms.

Solution Approaches to VRPs. As noted in the opening paragraph of this section, VRPs are among the most rewarding (and difficult) of all problems to solve, and much has been published in recent years on solution techniques. Ronen identified four common approaches to solving vehicle routing problems – manual, pure optimization (exact), optimization with embedded heuristics, and pure heuristics [Ronen, 1988:141]. Because the manual and pure heuristic approaches depend heavily on the specific application, the analysts who use these approaches do not normally publish their work in technical journals. Consequently most of the literature deals with the "exact" and "optimization with embedded heuristics" approaches.

Optimal solutions can be found to small problems by using direct tree search methods, dynamic programming, or integer programming [Laporte, 1992:346]. Unfortunately, the largest problem that can be solved using these methods is still quite small. Most routing and scheduling problems of interest are NP-hard. NP-hard problems are a class of network and combinatorial problems for which no polynomially-bounded solution algorithm has yet been found (a polynomially-bounded algorithm is one whose computational burden increases only polynomially in the worst case as the problem size increases). Because the VRP class of problems is NP-hard, they become difficult to solve as the number of vehicles and customers increases, so exact solution approaches can only be used on small, simple problems. The largest vehicle routing problem with time windows solved using exact methods until recently invoived only 4 vehicles and 14 customers. However, recent progress has increased the size of solvable prob!ems to about 100 customers by decomposing the problem and using a combination of exact methods [Desrochers et al., 1992:342].

Most problems of practical size are solved using heuristics or by a combination of optimization and heuristic methods. "A heuristic algorithm is a procedure that uses the problem structure in a mathematical (and usually intuitive) way to provide feasible or near-optimal solutions" [Bodin et al., 1983:77]. A heuristic is considered effective if the solutions it provides are consistently close to optimal. Most VRP heuristics fall into three broad categories – tour construction procedures, tour improvement procedures, and composite procedures [Bodin et al., 1983: 87]. Linear programming (LP) can also be thought of as a heuristic algorithm for VRPs. Relaxation of the requirement for an integer solution greatly increases the size of the problems that can be solved. However, the resulting non-integer solutions may have very limited meaning and may not resemble the optimal integer solution very closely.

Heuristics are used extensively to solve real-world problems because of the limitations of exact methods, but their performance depends heavily upon the particular application. While heuristics generally provide a "good" feasible solution, it is often difficult to determine how close the heuristic solution is to the optimal solution. Consequently, the exact methods are preferred when the problem is small enough that an optimal solution can be found in a reasonable amount of time.

Application to the ACEP Model. Since the ACEP model represents an instantiation of a potentially large vehicle routing problem, a heuristic technique for obtaining a solution appears to be the best alternative. The ACEP model developed by Busch and Hilliard can be considered a heuristic solution method for two reasons:

(1) Linear programming is used as a method of obtaining an optimal solution, but the resulting solution contains non-integer values and is not feasible without further processing. This may prove to be adequate only if very aggregate results are required.

(2) The computationally difficult problem of determining the optimal routing for each aircraft is largely avoided by including only the most practical routes. This is

acceptable in an airlift situation where only a very limited number of airfields are available and the routes can be easily enumerated.

Certainly other heuristic algorithms could be developed that would provide feasible (i.e. integer) solutions. One goal of the validation process is to determine whether or not the heuristic technique chosen is adequate. For this reason, it is important to validate any VRP formulation before using the model to support routing and scheduling decisions. The next section provides an overview of verification and validation techniques commonly used in both operations research and software engineering.

Verification and Validation Techniques

Clayton Thomas, a former Chief Scientist of the Air Force, once said that "all models are wrong, some are useful". Models are wrong because they are an inexact representation of some real-world system or problem. Determining whether or not a model is "useful" is the goal of the validation process. The requirement to validate models is common to all engineering activities, but has received remarkably little attention in most fields of engineering. However, a relatively mature validation paradigm (model) has developed in the field of software engineering over the last decade. This paradigm can be applied to model development as well.

Computer Software Validation. The development of mathematical models and the development of software systems have many parallels. Both represent an abstraction of a real system or problem. Much progress has been made in recent years in developing a structured method for the verification and validation (V & V) of computer software. This progress was made possible largely as a result of the recognition of the "life cycle" process of software development.

The classic software development life cycle is presented in Figure 3. The output of each phase becomes the input to the next; and the development process becomes a controlled transformation of the system requirements to software design, to software modules (computer code), and finally to an executable system. This life cycle, also called the "waterfall" life cycle, developed as a natural consequence of the need to control the transformation of the user's

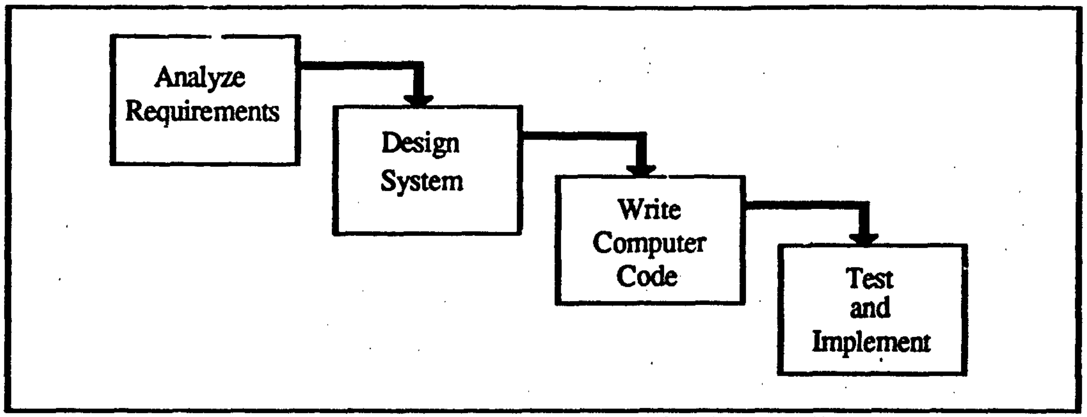


Figure 3. Classic Software Life Cycle Paradigm

requirements for the system into executable computer code. At each stage of the process, approximations and simplifying assumptions are made in order to "model" the previous stage. Consequently, flaws can be introduced at each stage which will cascade through the subsequent stages if no attempt is made to find and correct them. These flaws fall into two general categories – errors made in defining the requirements for the system, and errors introduced in transforming the system from one stage to the next (e.g. transforming written requirements into a system design). *Validation* is then defined to be the process of identifying and correcting the first type of error – errors in the requirements, and *verification* is the process of ensuring each transformation from one stage to the next is correct.

Historically, the primary method of performing software V & V has been post-development testing. Testing is the process of identifying discrepancies between actual results and expected results [Principles of Testing, 1985:3-1]. Since discrepancies (flaws) may be introduced at each stage of the life cycle process, V & V techniques must be able to find the flaws and identify the stage where each flaw was introduced. The primary disadvantage to post-development testing (testing after the system is built) is that flaws introduced early in the life cycle cascade through subsequent stages and become difficult and expensive to find and correct. For example, a flaw made in defining the user's requirements for a system can be 60 to 100 times more costly to correct after the system is built than during the requirements analysis stage [Pressman, 1987:17].

In recent years a more structured approach to software validation has been developed. More emphasis is placed on the early stages, especially the analysis of requirements. Testing is performed after each stage of the life cycle to verify and validate the results before continuing to the next stage. In this way, the most difficult and costly flaws (the flaws introduced in the early stages – such as the requirements analysis) are identified and corrected before they can effect subsequent stages. The transformation of one stage to the next is further controlled through documentation of the process (Figure 4).

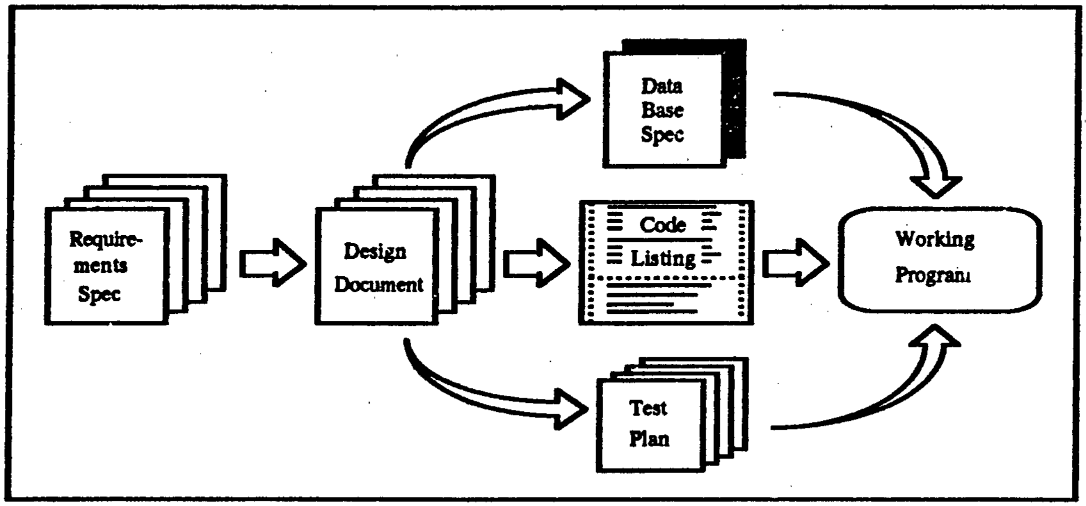


Figure 4. Software Life Cycle Documents (Pressman, 1987:18)

A prerequisite to finding flaws in system requirements is to obtain a written "requirements specification" which acts as a contract between the software developer and the software user. This specification defines the scope of the software system for the developer and helps define the boundaries of the system. It also defines the major functions and output expected of the system, helping the user realize the system's capabilities and limitations before it is built. The requirements specification is written at the user's level without software engineering jargon which might obscure the intent [General Electric, 1986:4-7]

After the requirements specification is completed and approved by the user, a system is designed to meet the specifications. The design is documented so that it may be verified against the requirements specification. Computer code is then written to implement the approved design. The final stage of testing is performed by the user of the system and is designed primarily to find any flaws in the specification that may still remain. Thus, the cornerstone of the validation process is the requirements specification. Another technique which has gained favor in recent years, and is made possible by the life cycle process, is the concept of independent testing.

Most operational modeling is performed by a team of one or more analysts who develop the model formulation, implement the model, and validate the results. For many years, this is how most computer software was developed as well. Recently, however, many software engineering organizations have formed independent test teams. "Independent testing is a cost-effective technique for finding flaws in software, and it is evolving as the standard method for verifying production application software" [Principles of Testing, 1985:4-5]. There are many benefits to performing independent testing [Principles of Testing, 1985:4-6]:

- The testing is conducted by personnel who have not been involved in the development of the software and can be more objective about the product and more aggressive in finding flaws.
- Requirements are reviewed from a different perspective, providing a valuable double check on the developer's interpretation.
- A separate test team is likely to be more critical in its interpretation of test results.

In summary, two important techniques can be borrowed from software engineering in performing validation tests on a mathematical model.

(1) A life cycle approach to model development may help guide the transformation of the user's requirements into a valid model.

(2) Final validation testing should be performed independently of model development whenever possible.

Unfortunately, the disciplined and widely accepted validation paradigm of the software engineering world has no parallel in the modeling world. Instead, a hodgepodge of post-development validation techniques are used depending upon the model form and the specific application. Consequently, the validation of models can be a more difficult task.

Mathematical Model Validation. A model, like computer software, is an abstract representation of some real-world problem. Approximations and simplifying assumptions are generally required to make the model tractable (capable of being solved). Model validation can be defined as the analytic process of proving that a model adequately represents the problem, and that a solution to the model is also a solution to the real-world problem. Hillier and Lieberman judge the validity of a mathematical model by "whether or not the model predicts the relative effects of the alternative courses of action with sufficient accuracy to permit sound decisions" [Hillier and Lieberman, 1986:20].

The development of large mathematical models requires a life cycle approach similar to software engineering. In *Model Building in Mathematical Programming*, H.P.Williams describes the process of building and validating a model as "a two-way process gradually converging on a more and more accurate representation of the situation being modelled" [Williams, 1985:96]. Figure 5 shows perhaps the most widely used model development life cycle. A *Conceptual Model* is the model builder's understanding of the important parameters, processes and interactions in the problem or system to be modeled [Alink and Blackstone, 1992: H-7]. An *Operational Model* is the implementation of the conceptual model into an executable form [Alink and Blackstone, 1992: H-7]. A *Valid Model* is an operational model that has been proven to adequately represent the problem for the intended use of the model.

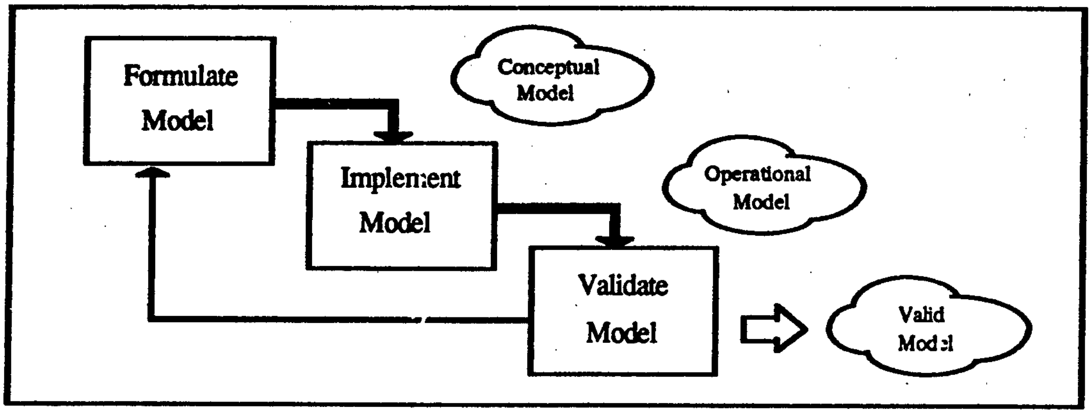


Figure 5. Traditional Model Development Paradigm

Only recently, however, has serious research begun on many of the issues associated with the life cycle approach to modeling, such as documentation standards and configuration management.

Many textbooks on operations research offer suggestions on how to validate mathematical models, and much has been written on the validation of other types of models, such as simulation models. Most of the techniques offered are post-development tests for validity. A survey of operations research, simulation, and expert system research into validation yields the following techniques:

(1) Face Validation. This technique involves having potential experts and people knowledgeable in the domain of the application examine the model in action and assess its performance at face value [O'Keefe et al., 1988:86].

(2) Constraint Validation. In many linear programming models, the objective function and constraints can be interchanged to provide additional insights into the validity of the model formulation. "It is often desirable to solve the model a number of times with different (possibly contrived) objectives in order to test out as many constraints as possible" [Williams, 1985: 96].

(3) Predictive or retrospective tests. When possible, historical data can be used as input to the model. A comparison of the model's solution to what actually happened may indicate whether using the model is a significant improvement over current practices. The technique is best described by Bazaraa, Jarvis and Sherali in *Linear Programming and Network Flows* as follows:

> The fourth stage [of model building] is *model testing, analysis, and (possibly) restructuring.* One examines the model solution and its sensitivity to various system parameters, and studies its predictions to various what-if types of scenarios. This analysis provides insights into the system. One can also use this analysis to ascertain the reliability of the model by comparing the predicted outcomes with the expected outcomes, using either past experience or conducting the test retroactively using historical data. [Bazaraa et al., 1990:8]

There are two disadvantages to this approach. First, it may use the same data that guided the formulation of the model [Hillier and Lieberman, 1986:23]. Second, the outcome of past events is only one of many possible outcomes, and it is unlikely that a model can incorporate all of the determining factors [Pritsker, 1986:13].

(4) Event validity or sensitivity analysis. Sensitivity analysis is performed by systematically changing the input parameters over some range of interest and observing the effect upon system performance [O'Keefe et al., 1988:86].

(5) Turing tests. While more commonly used to validate expert systems, Turing tests can be used to validate any kind of model designed to replace a previous method. The test is conducted by providing experts with output from both the model and the previous method without knowing the origin of each set of output. If the experts cannot tell the difference, then the test is a success. The Turing test is especially helpful in establishing the validity of a new system where the users are reluctant or skeptical [O'Keefe et al., 1988:86].

(6) Field tests. As a last resort in validation techniques, the model can be placed in operation to determine how well it performs. Normally, the previous method of obtaining solutions continues to be used as well to provide an additional tool for evaluation [O'Keefe et al., 1988:86].

These validation techniques are widely used by analysts, but very little guidance is published on how to perform each technique since their application is highly problem dependent. In addition, each of the techniques is a post-development test for validation in that the model must be created before they can be used. Thus these techniques are only really effective when they are coupled with a strong life cycle development approach.

In summary, a review of the literature leads to three conclusions about the ACEP model validation problem. First, since the problem represents an instantiation of a large vehicle routing and scheduling problem, a heuristic technique for obtaining a solution is necessary. One of the tasks of the validation process is to determine if the heuristic technique used is adequate. Second, a review of validation techniques in both software engineering and operations research indicate that validation of mathematical models may be best accomplished through a systematic life cycle development approach coupled with post-development validation tests. And finally, there may be advantages to performing the validation independently of model development.

In the next chapter, a new model development life cycle is proposed and applied to the ACEP model to begin the validation process.

III. *Life Cycle Validation Approach*

The model validation approach used on the ACEP model is based upon the validation paradigm of software engineering reviewed in the previous chapter. A disciplined and well documented life cycle development approach was found to be the most effective way to validate a model. In this chapter, a model development life cycle is proposed and applied to the ACEP model. As a result, significant flaws are found in the original ACEP design.

The first step in the validation process is to define a general model development life cycle. The difficulty in applying a life cycle validation approach to the ACEP model is that a model design has already been proposed while no formal written requirements for the model are available. Thus, the second step is to recover the undocumented life cycle stages already completed on the ACEP.

Proposed Model Development Life Cycle

The proposed model development life cycle can be described as a four step process similar to the classic "waterfall" life cycle used in software development (Figure 6). Each stage is

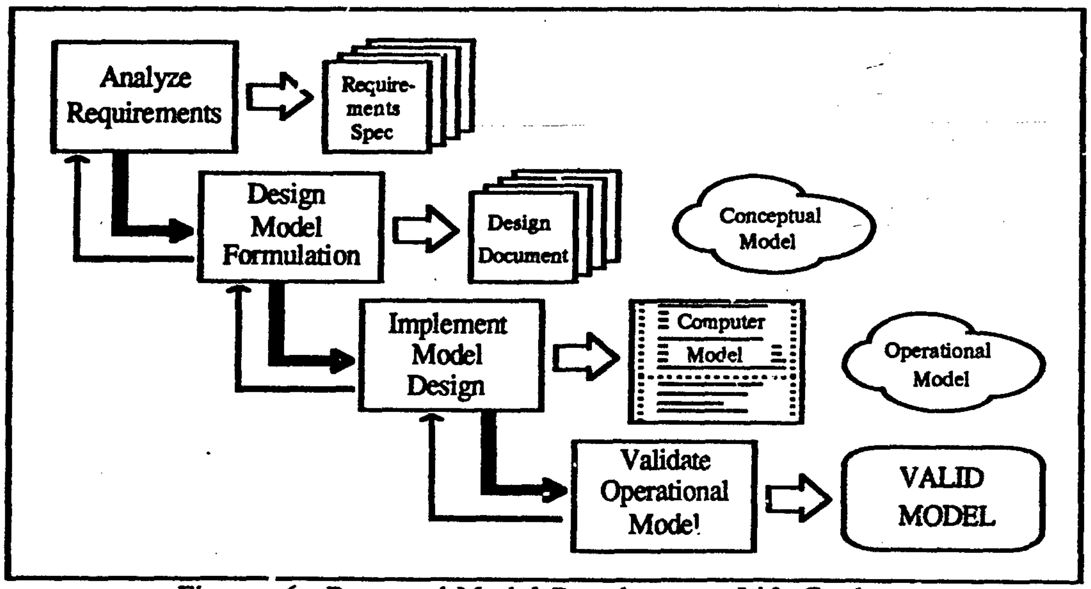
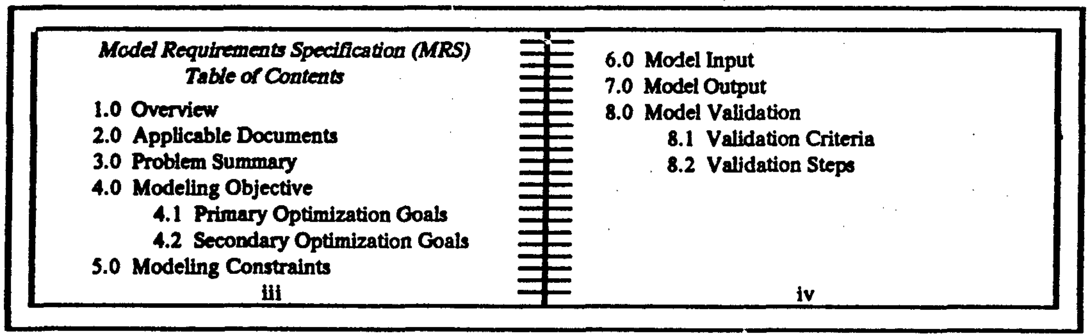


Figure 6. Proposed Model Development Life Cycle

dependent upon the previous stage for its requisite input. The feedback lines in the diagram indicate that the process is iterative. When a flaw is discovered in one stage, the process is backtracked to the stage where the flaw was introduced and restarted. Most model developers use a rough approximation of this life cycle naturally. However, a more disciplined use of the process is neede℃ to develop large, complex models and when a model is to be used more than once. In addition, formal documentation of each stage of the life cycle process may help greatly in the use and maintenance of the model.

Analysis of Requirements. The first and most important stage of the model development life cycle is the analysis of requirements. The output of this stage is a formal, written requirements specification. This document puts in writing the requirements of the model, including a description of the problem being modeled, thc expected inputs to, and outputs from the model, and the performance criteria that the model is expected to meet. The content of this document should also include the motivation for the model, the intended use of the model, and the specific post-development steps planned to validate the model for operational use. The requirements specification should not (in theory) be constrained to any particular modeling approach or solution methodology, except where the users of the model are constrained by available modeling resources. A typical model requirements specification might include the items shown in Figure 7.



Model Requirements Specification (MRS)
Table of Contents

1.0 Overview
2.0 Applicable Documents
3.0 Problem Summary
4.0 Modeling Objective
4.1 Primary Optimization Goals
4.2 Secondary Optimization Goals
5.0 Modeling Constraints

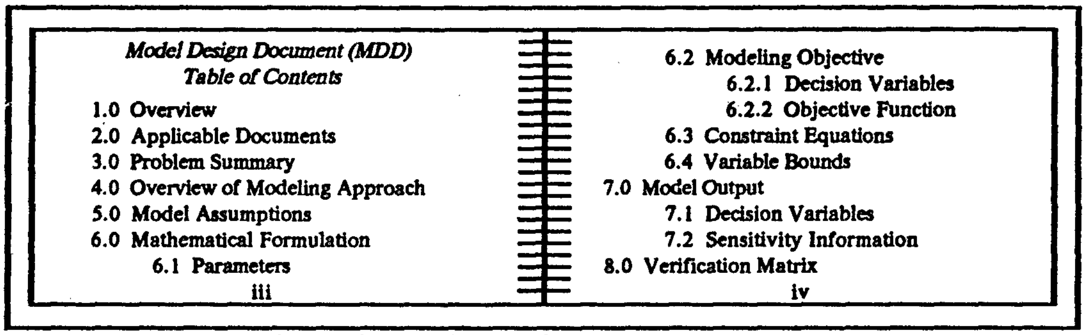
iii

6.0 Model Input
7.0 Model Output
8.0 Model Validation
8.1 Validation Criteria
8.2 Validation Steps

iv

Figure 7. Model Requirements Specification Format

Model Design. The second stage is to transform the requirements into a model design (the "conceptual model"). The design document should describe the modeling approach taken (e.g. linear programming), the assumptions required to use the modeling approach chosen, and the model design. While Data Flow Diagrams and Data Dictionaries have become the standard design tools in software engineering, no such standard design tool has emerged in modeling. Thus, a wide variety of design tools can be used. In simulation models, for example, the model design might be represented by network flow diagrams. In mathematical models such as the ACEP model, the design is most often represented by a mathematical formulation comprised of parameters, variables, and equations which relate the parameters and variables to each other. Since a proposed ACEP formulation already exists, the design document in this case expands upon the formulation and updates the model based upon the "recovered" model requirements. Figure 8 shows the format of the design document used for ACEP.



Model Design Document (MDD)
Table of Contents

1.0 Overview
2.0 Applicable Documents
3.0 Problem Summary
4.0 Overview of Modeling Approach
5.0 Model Assumptions
6.0 Mathematical Formulation
6.1 Parameters
iii

6.2 Modeling Objective
6.2.1 Decision Variables
6.2.2 Objective Function
6.3 Constraint Equations
6.4 Variable Bounds
7.0 Model Output
7.1 Decision Variables
7.2 Sensitivity Information
8.0 Verification Matrix
iv

Figure 8. Model Design Document Format

Model Implementation. The third stage is the transformation of the design into an executable (or "operational") model. This is normally done with a mathematical modeling system or by writing computer code. In some cases, the transformation from design to implementation is automated. In any case, the transformation is required to produce a model which generates the desired information. The main validation goal of this stage is to ensure the operational model correctly implements the conceptual model. Each constraint is examined one at a time in a logical order. The effect of each type of constraint on the solution of a contrived set of input data is compared with what is expected, and any discrepancies are investigated. This process is called *constraint validation*. The ACEP design was implemented using the General Algebraic Modeling System (GAMS) and the constraint validation process used to verify the GAMS model is outlined in Chapter IV.

Operational Validation. The final stage in this transformation is a post-development validation process to ensure the executable model accurately represents the problem, meets the performance and validation criteria outlined in the requirements specification, and is suitable for the intended use of the model. This process can be called "operational validation" [Alink and Blackstone, 1992: H-8]. The retrospective test described in the literature review is used in this step and is described in detail in Chapter V.

The proposed life cycle for model development offers two major advantages over the traditional method of model building. First, the model formulation is divided into two distinct phases – analysis of requirements and model design. This places more emphasis on the investigation of requirements in the beginning which increases the chances that the model developed will be adequate for its intended use. Second, the results of each stage of development are documented. This allows for a "face validation" of the model earlier and makes an independent evaluation of the model's validity possible.

Clearly this proposed life cycle has merit in the development of new models. However, the ACEP model already exists in a conceptual form. The next section describes the process used to apply the life cycle to the existing ACEP model design.

Recovery of Model Requirements

The ACEP model proposed by Busch and Hilliard describes a conceptual model design. To recover the missing life cycle stages, that portion of the normal development life cycle that lies "upstream" from the existing model has to be recreated. In the case of the ACEP model, the requirements for the model have to be analyzed and a written specification developed before the design can be implemented (Figure 9). Three approaches are possible in reconstructing model requirements: starting over, "backing in", or a combination of both.

(1) Starting Over. The existing design can be ignored and the entire life cycle process started over. The advantage to this approach is that any modeling approach can be taken and great improvements in the final model are possible. This approach is likely to take more time and effort, however, and ignores the contribution of previous work that was performed to create the existing design.

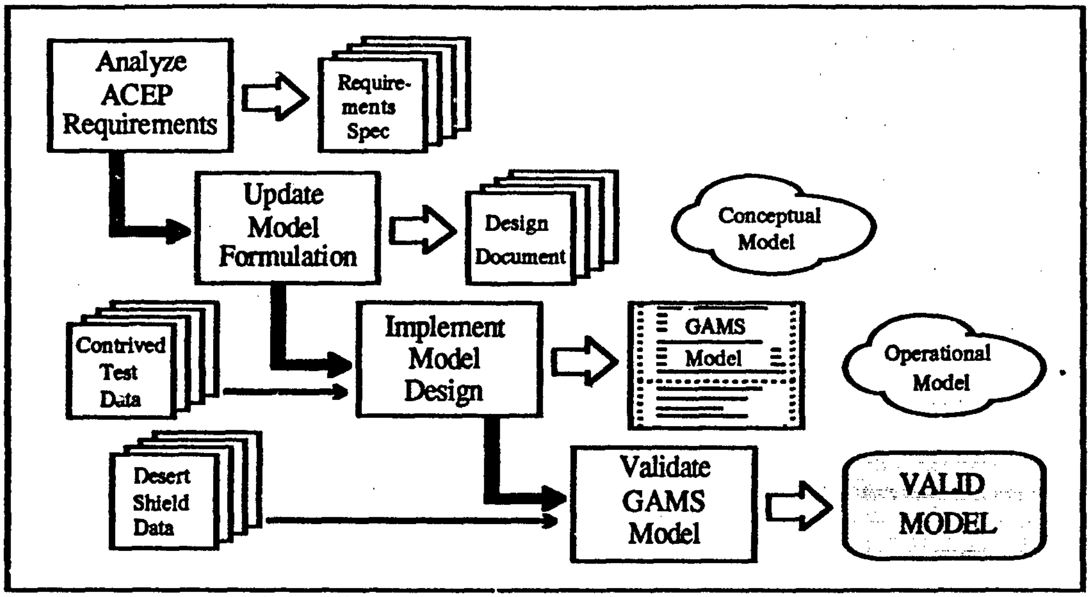


Figure 9. ACEP Model Validation Process

(2) Backing In. Using a variation of Meyer's "backing in" approach, the requirements specification can be created to lead directly to the existing design [Meyer, 1987]. The model user's approval of the requirements specification "as is" then represents compelling evidence that the existing design is adequate. Similarly, changes to the specification requested by the model user should lead directly to improvements in the design. This approach might be the best alternative when the requirements are not well understood, when the existing design is likely to be adequate, or when time and effort constraints are imposed that prevent a complete rework of the model.

(3) Combination of both approaches. A combination of approaches can be used when some requirement changes are known ahead of time. This approach was used to recover the requirements for the ACEP model. Most of the requirements in the specification were described in such a way as to lead more or less to the representation of the requirement used in the model design ("backing in"). However, some additional model requirements were evident from the beginning and were included in the first draft of the requirements.

The ACEP Model Requirements Specification (MRS), contained in Appendix A, was developed through research into the airlift planning problem, analysis of the ACEP design, and numerous interviews with analysts at Air Mobility Command. Each iteration of the specification was reviewed by the model sponsor as well as by other experts on the airlift planning problem, providing a valuable "face validation" of the model's requirements.

ACEP Design Changes

Once the recovered requirements specification was approved by the model sponsor at Air Mobility Command, then the model design was updated to reflect the new requirements. The ACEP Model Design Document (Appendix B) was created based upon the original ACEP design and the new model requirements. Changes to the design which resulted from the new model requirements include:

(1) Route structure. The original design assumed that the off-load airfield was the last stop on each route, failing to account for the return of the aircraft to their home base. A recovery base near the off-load airfield was also added to the route structure, although this new requirement did not effect the current model design (see Appendix A, Section 5.3 for a more detailed description of the route structure).

(2) Working MOG. The original design included flow constraints to account for the available ramp space at each airfield. However, the analysts at AMC normally work with two different kinds of MOG (maximum on the ground) in planning airlift operations – parking MOG which accounts for ramp space, and a "working" MOG which accounts for other factors such as refueling capability (see Appendix A, Section 5.4 for a more detailed description of MOG).

(3) Minimum load requirement. The new design includes the option to specify a lower bound on the number of missions scheduled, preventing the model from scheduling aircraft with loads below a certain percentage of capacity (see Appendix A, Section 5.5)

(4) Aircraft utilization constraints. The utilization of airlift aircraft is constrained by a number of factors including crew limitations and maintenance requirements. Air Force planners aggregate these factors into a utilization rate (called UTE). The Busch and Hilliard formulation of ACEP did not include UTE constraints, but UTE was included in the requirements at the request of the model sponsor. The aircraft utilization

constraints required additional input to the model including information on the expected flight time between airfields on each route, the expected ground time at each stop, and the objective UTE rate for each aircraft and surge perioa (see Appendix A, Section 5.5).

To summarize, the proposed model development life cycle offers two major advantages over the traditional model development paradigm: (1) more emphasis is placed on investigating the needs of the model sponsor, increasing the chances that the model developed will be "useful"; and (2) the transformation of the model through each stage of the life cycle process is documented to help in the validation process as well as in the use and maintenance of the model. Three approaches were discussed for applying the proposed life cycle to the existing ACEP model: (1) starting over; (2) "backing in" to the model requirements; and (3) a combination of (1) and (2). The combination approach was used to improve the existing design of the ACEP model.

In the next chapter, the process used to build and validate an executable version of the ACEP model design using GAMS is discussed.

IV. *Constraint Validation*

The recovery of model requirements provided a valuable tool in improving the Busch and Hilliard ACEP model design. The third stage in the proposed model life cycle is to build an operational model that implements the conceptual model design. In this chapter, the process used to build an executable version of ACEP using the GAMS modeling language is presented. In addition, constraint validation techniques for ensuring the operational model correctly implements the conceptual model design are discussed. Finally, the results of using these techniques on the ACEP are given.

GAMS Implementation

GAMS acts as a "front-end" and a "back-end" to a solver and is designed to make the formulation and maintenance of large and complex mathematical models easier [Brooke et al., 1988: Preface]. The GAMS modeling language provides an impressive array of tools for manipulating the input parameters as well as the solution and sensitivity information. Figure 10 shows the process GAMS uses to obtain a solution to a model. A special interface program must be written to pass the model from GAMS to a solver, but interface routines for the most widely used solvers are available. For the validation runs of the ACEP model, the MINOS solver (Version 5.2, March 1988) was used.

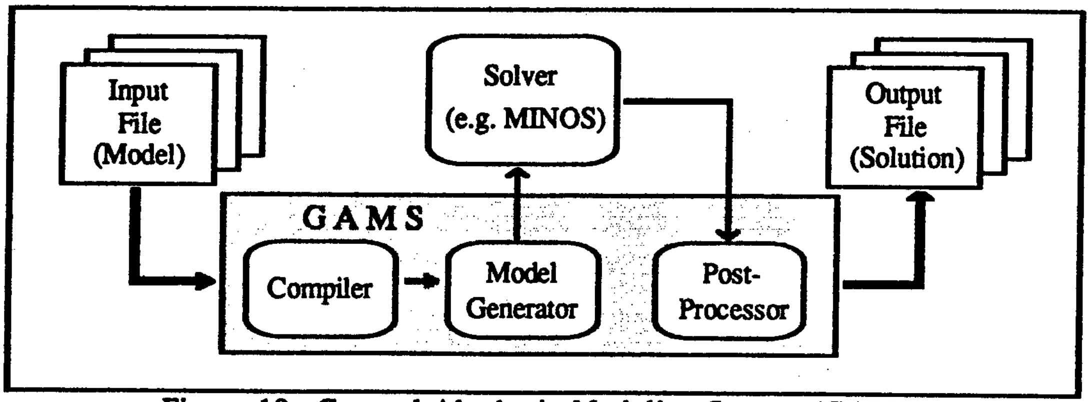


Figure 10. General Algebraic Modeling System (GAMS)

The GAMS implementation of ACEP, provided in Appendix C, represents yet another level of abstraction from the real problem. More assumptions and approximations are required to implement the model formulation. Consequently, the next step after building the operational model is to verify that it is an accurate implementation of the conceptual model. In doing so, further insight into the model and the problem is gained providing an additional tool for validation. This process is called *constraint validation* and has three main goals:

(1) Verify that each constraint works – that the mathematical representation of each resource is correctly implemented in the model.

(2) Ensure that the representation of each constrained resource does not cause unexpected "side effects" that result in unrealistic model solutions when the constraint is enforced.

(3) Provide further insight into the validity of the model requirements and design.

The process of constraint validation was accomplished in three primary ways: (1) through the development of a contrived airlift scenario designed to "stress" the model; (2) by systematically adding constraints to a skeletal model; and (3) by examining the solution to alternative objective functions.

Contrived Scenario. A designed set of input data is necessary to initially test the operational model and to act as a baseline scenario for further constraint testing. An airlift scenario was designed for ACEP to make each constraint binding at some point during the planning horizon of the model. This was possible because of the multi-time period aspect of the ACEP model. Without this aspect, many different scenarios would have to be created to accomplish the same objective. The primary resource constraints of interest in the ACEP model are:

(1) Airfields. The on-load and off-load airfields are limited in the amount of cargo and passengers that can be processed during each unit of time (cargo throughput and passenger throughput). All airfields have a limited amount of ramp space for parking aircraft (parking MOG). In addition, airfields may have limited capability to service aircraft (refuel, maintain, etc. – called working MOG). See Appendix A, Section 5.4 for a complete description of the airfield resource constraints, and Appendix B, Section 6.3.3 for the mathematical formulation of the constraints.

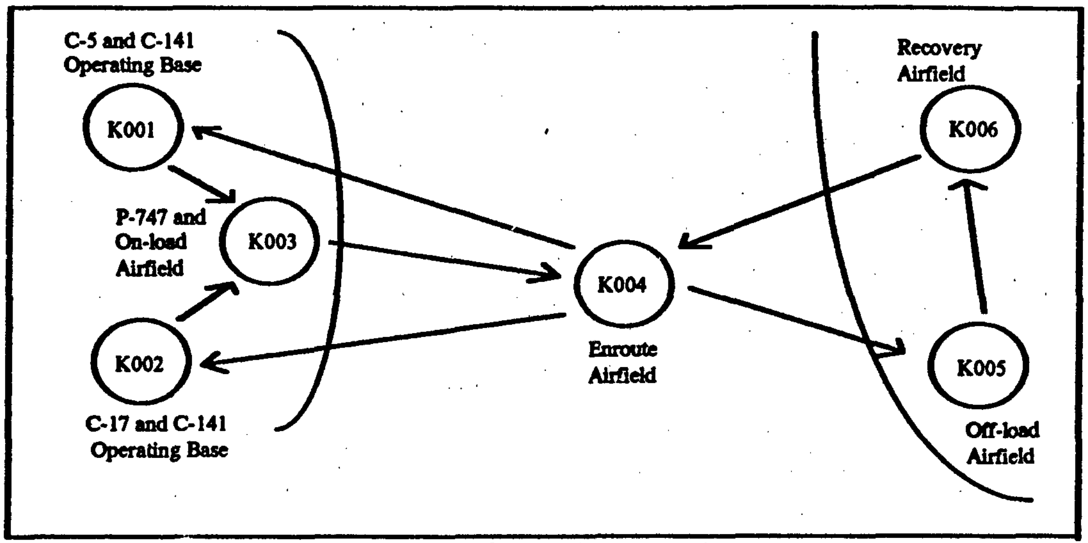


Figure 11. Test Scenario Route Structure

(2) Aircraft. A limited number of aircraft are available at any given time and the utilization (UTE) of the aircraft (number of flight hours accumulated over a given period) is limited. See Appendix A, Section 5.5, and Appendix B, Section 6.3.3 for more detail on the aircraft resource constraints.

A route structure was designed to force bottlenecks in the system (Figure 11). Table 1 lists the airfields with designed shortages in constrained resources and the time periods in the model when the shortages occur.

TABLE 1

Designed Scenario – Airfield Bottlenecks

Airfield	Resource Shortage	Time Periods
K003	Cargo Throughput (MHE)	C01 – C05
K005	Pax Throughput (Pax Terminal)	C01 – C08
K004	Working MOG (Fuel trucks)	C01 – C64
K006	Parking MOG (Ramp Space)	C01 – C64

Tests of the aircraft resource constraints (availability and utilization) were designed by enforcing low surge period UTE constraints during the first three days of the planning period and by modifying the penalty for using an aircraft. The penalty was modified to increase with time, penalizing any delays in delivering the cargo and passengers after the start of the pickup window.

This forced the model to schedule the aircraft at the maximum utilization and availability early in the model to deliver the cargo and passengers as soon as possible.

The model was executed with the test scenario data using the primary modeling objective of minimizing shortfall and closure (day of last delivery). The results showed that each of the constraints was working correctly and no unexpected "side effects" were immediately apparent. However, further refinements were made to the post-processing of the solution to provide more useful information. Of particular interest was the sensitivity analysis. While this information can be printed automatically by GAMS, the sheer volume of the information available in even a small scenario is overwhelming. The most useful sensitivity information was the marginal value of each of the constrained resources. The LP solver computes a marginal value for each constraint equation in the model. To aggregate this information, post-processing was added to gather (sum) the marginal cost information across the relevant time periods. For example, Table 2 shows the information computed on airfield MOG used in the optimal solution.

TABLE 2

Designed Scenario – MOG Sensitivity Information

Airfield	Maximum Percent of Capacity Used	Accumulated Marginal Cost
K001	56.0%	
K002	35.7%	
K003	93.0%	
K004	100.0%	30.887
K005	51.7%	
K006	100.0%	13.502

The table shows that the full capacity of the MOG resource was used at both airfield K004 and airfield K006 as expected. In addition, the marginal cost information indicates that the shortage of MOG at airfield K004 has a larger impact on the solution than K006. In fact, a one unit increase in the MOG available at K004 across the entire planning period will improve the objective function by up to 30.887 units. The actual improvement may be less, however, since the aggregation of the marginal cost across the planning period assumes that the current basis remains optimal after the change (the actual improvement in this case was 21.8 because the basis changes). The aggregate marginal cost provides a good way of determining which of the constrained resources offers the most potential improvement in the objective function by adding to the availability of the resource. Cnce a good contrived scenario has been designed and tested, it can be used in a more detailed analysis of each of the model constraints.

Constraint Examination. The designed scenario provides a good initial assessment of the operational model. The next step is tɔ perform a more detailed examination of each of the constrained resources. This is done by stripping the model of all but the most basic parameters and equations and then introducing the resource constraints one at a time to observe their individual effects. With four primary types of constrained resources (the number and utilization of aircraft, and the airfield MOG and throughput), there are 4! = 24 ways in which these four constraints can be introduced to the model. However, there are logical considerations that can be used to eliminate some combinations. For example, the UTE constraints in the ACEP model cannot be computed without providing the maximum number of aircraft available. Thus the UTE constraints must be added to the model after the aircraft availability constraints. In this manner infeasible combinations can be eliminated from consideration and one of the remaining combinations chosen to begin the test.

The basic ACEP model includes the objective function and the constraints which account for the delivery or shortfall of the cargo and passengers (DELIVPAX, DELIVBLK, and DELIVOUT). In addition, the NONPREF constraints are included in the basic model since they have no effect on resource consumption (see Appendix C). Table 3 shows the order in which the constraints were introduced to the basic model and the effect of each additional constraint on the size and density of the constraint matrix (percent of matrix elements that are nonzero).

TABLE 3

Constraint Validation – Model Size and Density
(Objective Function – Minimize Cost)

Model Characteristic	Base	MOG	Thruput	Avail	UTE
# of Constraints	72	444	684	932	944
# of Variables	1740	1740	1740	1740	1740
Density	2.9%	1.3%	1.6%	1.5%	1.6%

The MOG equations represent the largest block of constraints (372), but are also the least dense – a significant factor in the effort required to find the optimal solution. Table 4 shows the effect of each constraint on the optimal solution.

TABLE 4

Constraint Validation – Effects on Solution
(Objective Function – Minimize Cost)

Solution Characteristic	Base	MOG	Thruput	Avail	UTE
# of Iterations *	325	273	1335	4209	3618
Objective Function Value	1,497.17	2,362.65	2,514.74	2,923.04	3,336.43
Latest Delivery Date	Day 29	Day 33	Day 35	Day 41	Day 47
Total # of Sorties	197	177	177	177	177
Sortie Mixture (Percent of Total by Aircraft Type)					
C-141	45.9%	33.5%	33.5%	33.5%	33.5%
C-5	34.4%	38.3%	38.3%	38.3%	38.3%
C-17	11.2%	12.4%	12.4%	12.4%	12.4%
P-747	8.5%	15.8%	15.8%	15.8%	15.8%
Cargo Mixture (Percent of total delivered)					
Out - C-5	100%	100%	100%	100%	100%
Blk - C-5	75.4%	75.4%	75.4%	75.4%	75.4%
C-17	24.6%	24.6%	24.6%	24.6%	24.6%
Pax - C-141	52.2%	34.2%	34.2%	34.2%	34.2%
C-5	20.5%	20.5%	20.5%	20.5%	20.5%
P-747	27.3%	45.3%	45.3%	45.3%	45.3%

* Represents the number of iterations performed by MINOS 5.2 to find an optimal solution starting from the optimal basis of the previous model.

Since the airlift scenario was designed to make each of the resource constraints binding, the objective function "cost" of the airlift as well as the closure (day of last delivery) is expected to increase as each constraint is added. It is interesting to note that the number of sorties, the percentage of sorties flown by each aircraft type, and the percentage of cargo/pax carried by each aircraft type does not change after the first constraint (MOG) is added. Further investigation revealed that these attributes of the solution are relatively insensitive to constraints and depend mostly on the number and type of aircraft employed, the movement requirements, and the time windows for pickup of the requirements – all of which are input parameters to the model.

Table 5 contains the usage information for each constrained resource, providing further evidence that the resource constraint equations are functioning correctly. However, more detailed analysis of the C-141 utilization information uncovered a flaw. In the designed scenario, the

TABLE 5

Constraint Validation – Maximum Resource Levels
(Objective Function – Minimize Cost)

Constrained Resource	Base	MOG	Thruput	Avail	UTE
Maximum MOG used (percent of capacity)					
K001	591%	68.2%	50.8%	52.2%	56.0%
K002	106%	60.0%	46.8%	33.6%	35.7%
K003	1,161%	100%	100%	86.3%	93.0%
K004	696%	100%	100%	100%	100%
K005	696%	60%	45.2%	47.7%	51.7%
K006	871%	100%	100%	100%	100%
Maximum Cargo Throughput (percent of capacity)					
K003	2,167%	211%	100%	100%	100%
K005	867%	127%	60.0%	60.0%	60.0%
Maximum Passenger Throughput (percent of capacity)					
K003	548%	78.0%	41.6%	48.4%	50.8%
K005	1,370%	195%	100%	100%	100%
Aircraft Used (percent of available)					
C-141	904%	485%	268%	100%	100%
C-5	287%	233%	150%	100%	100%
C-17	314%	298%	158%	100%	100%
P-747	523%	318%	316%	100%	100%
Surge UTE Rates Achieved (percent of maximum)					
C-141	N/A	N/A	N/A	298%	100%
C-5	N/A	N/A	N/A	151%	100%
C-17	N/A	N/A	N/A	171%	100%
P-747	N/A	N/A	N/A	138%	100%

C-141 aircraft are based out of two different airfields (K001 and K002). When the availability and UTE levels are computed separately for each group of C-141s, as in Table 6, two problems become evident. First, the availability constraint is enforced across aircraft types, allowing the model to use more aircraft than are available from each operating base. This occurred at both of the C-141 bases at some time during the planning period (though not at the same time). Similarly, the UTE constraints are enforced over an aircraft type, allowing the model to over-utilize the aircraft from an advantageously located base (K002) while under-utilizing the aircraft from a base farther from the on-load airfields (K001) to maintain the required overall UTE rates. Consequently, the ACEP model was changed to enforce both availability and UTE by operating base as well as aircraft type.

TABLE 6

Analysis of Optimal C-141 Utilization by Operating Base
(Objective Function – Minimize Cost)

Aircraft/Operating Base	Availability	UTE
C-141 (Overall)	100.0%	100.0%
C-141 (K001)	150.0%	38.5%
C-141 (K002)	174.0%	223.0%

Alternative Objectives. The final step in the constraint validation process is to repeat the process of examining the constraints using alternative objective functions. Often one of the resource constraints can be used as an objective function, with the goal of minimizing the use of the resource subject to meeting a set value of the original objective. The goal in switching the constraints and objective function is to learn more about the behavior of the model and correct any inappropriate behavior.

For this step of the validation process the objective function chosen is to maximize the flow of cargo and passengers, where the objective function value reported is the maximum number of passengers and tons of cargo which can be delivered over the planning period. To use this objective function, it is assumed that an infinite amount of cargo and passengers is available for pickup at each on-load airfield, but the pickup windows are not changed. Table 7 shows the results of the model run with the new objective function.

TABLE 7

Alternative Objective – Effects on Solution
(Objective Function – Maximize Flow)

Solution Characteristics	Base	MOG	UTE	Thruput
# of Iterations *	380	1939	882	2648
Objective Function Value	64,908.00	60,667.00	52,023.18	50,023.18
Total # of Sorties	498	399	402	402
Amount Delivered (by class)				
Out (tons)	7,560.00	7,376.65	5,584.43	5,425.02
Blk (tons)	4,200.00	349.18	3,361.77	3,521.19
Passengers	53,148	52,941	43,077	43,077

* Represents the number of iterations performed by MINOS 5.2 to find an optimal solution starting from the optimal basis of the previous model.

Since the resource constraints act to limit the flow through the airlift system, the objective function is expected to decrease as each constraint is added. However, it is also evident that the model tends to favor the delivery of passengers over cargo. The average capacity of the passenger-capable aircraft (C-141 and P-747 in the scenario) is about 250 passengers, while the average capacity of the cargo-capable aircraft (C-5, C-17, and C-141) is about 40 tons. Since one passenger and one ton of cargo carry equal weight in the objective function, the model will naturally attempt to schedule as many of the higher capacity passenger aircraft as possible. However, Air Force planners use a rule-of-thumb that one ton of cargo must be delivered for each passenger [Litko, 1992]. This rule-of-thumb has proven to be roughly accurate in past airlift operations including Desert Shield. This ratio is true for the operation as a whole, but is not necessarily the case for each deploying unit.

There are a number of ways to indirectly cause the model to deliver roughly equal amounts of cargo and passengers. First, the penalty for shortfall in the cargo categories can be adjusted to compensate for the difference in the carrying capacities of the aircraft. Second, the objective function can be formulated to penalize the use of aircraft to carry passengers. However, further experimentation with these solutions showed that they are only effective when the objective is to minimize the shortfall and the airlift system does not have the capacity to deliver all the cargo that must be moved.

The best way to directly influence the ratio of cargo/passengers delivered is simply to add a constraint to the model which forces the shortfall in cargo to equal the shortfall in passengers within some specified tolerance. In the case where the objective is to maximize the flow, then the constraint will force the tons of cargo delivered to roughly equal the number of passengers delivered. A single new constraint (RATIO) was added to the ACEP model to force the tons of cargo (bulk and outsize) delivered to be within 10 percent of the number of passengers. This change was made to the model and the results are shown in Table 8. The effect on the solution is dramatic, but results in a more realistic delivery of cargo and passengers. The primary "side effect" of the new constraint in the test scenario is that the C-141 aircraft previously used exclusively for carrying passengers are now used only for carrying bulk cargo.

TABLE 8

Alternative Objective – Effects on Solution #2
(Objective Function – Maximize Flow)

Solution Characteristics		Base	MOG	UTE	Thruput	Ratio
# of Iterations *		380	1939	882	2648	634
Objective Function Value		64,908.00	60,667.00	52,023.18	50,023.18	22,014.08
Total # of Sorties		496	399	402	402	352
Amount Delivered (by class)						
Out (tons)		7,560.00	7,376.65	5,584.43	5,425.02	4556.09
Blk (tons)		4,200.00	349.18	3,361.77	3,521.19	5871.63
Passengers		53,148	52,941	43,077	43,077	11,586
Cargo Mixture (percent of total delivered)						
Out -	C-5	100%	100%	100%	100%	81.4%
	C-17					18.6%
Blk -	C-141					62.2%
	C-5		2%		4.6%	
	C-17	100%	98%	100%	95.4%	37.8%
Pax -	C-141	55.0%	55.3%	58.1%	58.1%	
	C-5	16.1%	15.8%	14.7%	14.7%	36.3%
	P-747	28.8%	29.0%	27.2%	27.2%	63.7%
Sortie Mixture (percent of total by aircraft type)						
C-141		45.2%	56.4%	47.9%	47.9%	54.6%
C-5		25.3%	30.8%	23.2%	23.2%	17.5%
C-17		21.1%	2.2%	20.9%	20.9%	22.1%
P-747		8.4%	10.5%	8.0%	8.0%	5.7%

* Represents the number of iterations performed by MINOS 5.2 to find an optimal solution starting from the optimal basis of the previous model.

Finally, the effect of each constraint on the utilization of resources is given in Table 9. When the RATIO constraint is added to the model, the passenger-capable aircraft are utilized less or used to deliver bulk or outsize cargo and the passenger throughput levels decrease. Thus, the model behavior matches what is expected when the model is forced to deliver more cargo.

To summarize, the primary validation task in the model implementation stage of the life cycle is to verify that the operational model correctly implements the model design. Specifically, the goals of the constraint validation process are to verify that each constraint performs it primary task of constraining the use or consumption of the resource; to ensure that no unwanted "side effects" are caused by the constraint equation; and provide additional insight into the model behavior.

Three techniques were used to complete the process with each contributing significantly to the validation effort. First, a scenario was developed to "stress" the model by forcing all

TABLE 9

Alternative Objective – Maximum Resource Levels
(Objective Function – Maximize Flow)

Solution Characteristics	Base	MOG	UTE	Thruput	Ratio
Maximum MOG used (percent of capacity)					
K001	120%	68.7%	69.4%	65.4%	62.7%
K002	54%	33.88%	47.4%	53.4%	51.2%
K003	289%	100%	100%	100%	100%
K004	199%	100%	100%	100%	100%
K005	173%	60%	60%	60%	55.1%
K006	434%	100%	100%	100%	100%
Sustained UTE Rates Achieved (percent of maximum)					
C-141 (K001)	105%	105%	100%	100%	100%
C-141 (K002)	125%	125%	100%	100%	100%
C-5	135%	132%	100%	100%	66.4%
C-17	125%	10.4%	100%	100%	92.6%
P-747	131%	131%	100%	100%	62.9%
Maximum Cargo Throughput (percent of capacity)					
K003	733%	346%	250%	100%	100%
K005	293%	138%	135%	60.0%	60.0%
Maximum Passenger Throughput (percent of capacity)					
K003	105%	71.9%	67.4%	71.6%	46.3%
K005	263%	164%	73.2%	91.6%	38.6%

constraints to be binding. During the development of the contrived scenario, the penalty for using an aircraft in the objective function was modified to increase with time to minimize closure as well as shortfall. In addition, it was discovered that marginal cost information computed automatically on the constrained resources could be aggregated into a single number representing the relative importance of each resource.

Second, model constraints were stripped from the basic model and introduced one at a time to observe their individual and collective effects on the solution and constrained resources. It was discovered that aircraft availability and UTE must be enforced for each group of aircraft operating from the same airfield as well as across aircraft types.

Finally, the constraint examination process was repeated with an alternative objective function of maximizing the flow of cargo and passengers. It was discovered that the model tends naturally to favor the delivery of passengers over cargo. Consequently, a single constraint was added to force the model to deliver roughly equal amounts of cargo and passengers within a given tolerance.

The ACEP model is now fully "operational". The last step in the proposed model development life cycle is to perform one or more post-development validation tests to determine how the model performs with "live" data.

V. *Retrospective Validation Test*

As a final step in the validation process, the ACEP model is used to "predict" the airlift capability of the Operation Desert Shield airlift system. Since the lessons of the Desert Shield airlift are one of the driving forces behind the development of the ACEP model, it seems fitting that the validation process include an evaluation of the model performance using Desert Shield data. In addition, this airlift operation represents one of the largest and most difficult airlift operations ever undertaken, providing a good "stress" test for the model.

Validation Criteria

Specific validation criteria were established during the reconstruction of the model requirements and are documented in the ACEP Model Requirements Specification (Appendix A). The primary objective of the model is to "obtain reasonable estimates of the number and type of aircraft needed . . . and identify bottlenecks in the airlift system" (excerpt from ACEP Model Requirements Specification). Specifically, four aspects of the model were chosen as the most important features for validation purposes:

(1) Ease of use. Information that is required for input to the model must be easily obtainable from sources already available in the planning process and not require extensive transformations prior to input.

(2) Response time. The model must provide a solution to a problem of realistic size within a reasonable amount of time. While any quantification of this criteria is purely arbitrary, a reasonable goal may be to provide a solution to a 30-day planning problem within 30 minutes of CPU time on a VAX minicomputer.

(3) Accuracy. The model must be able to determine the resources required to perform an airlift flow within plus or minus 10 percent of the actual requirements. This criteria represents a compromise by the model sponsor in that a higher degree of accuracy, while certainly desirable, requires an unreasonable amount of effort in both obtaining input data with a corresponding degree of accuracy, and in finding an optimal solution.

(4) Output. The output from the model must clearly state the optimal resource requirements, expected shortfalls, the most valuable aircraft types, and identify the major airfield bottlenecks in the system.

The first two criteria – ease of use and response time – become very important when the model is used during execution planning, which is the more demanding use of the model. Note that the validation criteria are established during the requirements analysis stage of the life cycle and are chosen by the model user – not the model developer – to represent the most important goals of the model.

Scenario

At 0100 (Kuwait time) on 2 August 1990, three Iraqi Republican Guard divisions began a ground assault into the neighboring country of Kuwait. At the same time, special forces from Iraq attacked Kuwait City and the Amir's palace. By 1900 the same day, the country was all but lost and Iraqi forces began massing in an apparent threat to advance into Saudi Arabia. President Bush ordered the start of operation Desert Shield [Conduct of the Persian Gulf War, 1992: 1].

One of the primary military objectives of Desert Shield was to develop a defensive capability in the Persian Gulf region to deter Iraq from further attacks [Conduct of the Persian Gulf War, 1992: 40]. The Military Airlift Command (MAC – the predecessor to AMC) was faced with the

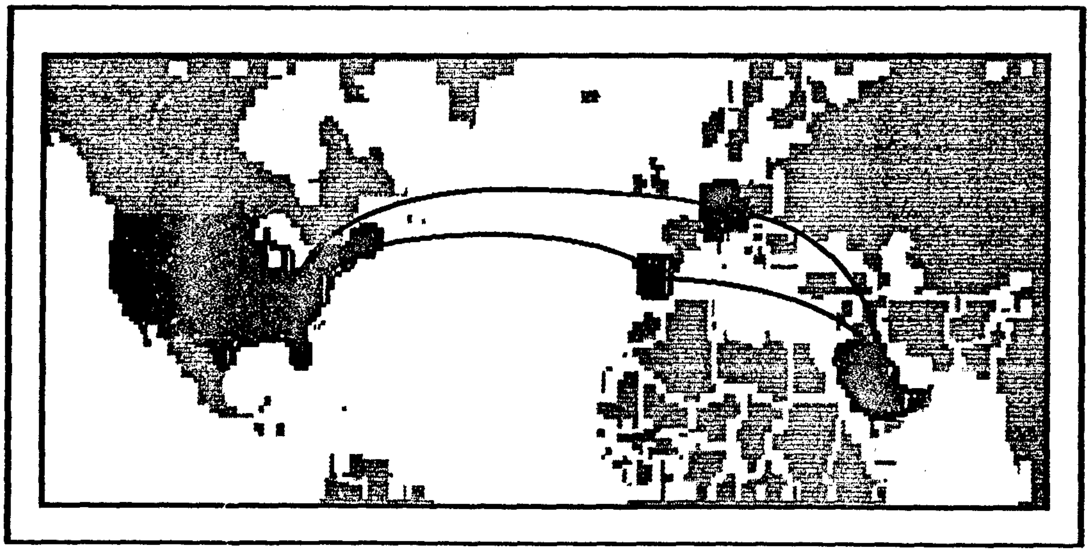


Figure 12. Overview of Desert Shield Airlift System

problem of moving an unprecedented military force halfway around the world in as short a time as possible. Figure 12 shows an overview of the airlift system on a regional map. Virtually half the fleet of strategic transport aircraft owned by MAC as well as civil reserve aircraft were used in the airlift. After six months of almost-round-the-clock airlift, more than 500,000 troops and 544,000 tons of cargo had been moved [Conduct of the Persian Gulf War, 1992: E-9].

The Desert Shield Model

The operational ACEP model is used to predict 30 days of Desert Shield operation. Since detailed records on the first 30 days are difficult to find, the input data represents the daily or monthly (as appropriate) average over the first 180 days of the operation, and only sustained UTE rates are enforced. Each time unit in the model represents one day in the airlift.

Airfield Resources. The ACEP model of the Desert Shield deployment uses 15 primary airfields. The on-load airfields are located in the continental United States and Europe. The off-load airfields are all located in Saudi Arabia. Table 10 lists the International Civil Aviation Organization (ICAO) designators, locations, capacities, and primary uses of the 15 airfields.

TABLE 10

Desert Shield Airfield Resources
(Source: AMC/XPYR)

ICAO			Maximum on	Throughput Capacity	
Designator	Location	Primary Use	the Ground	Cargo	Passengers
KDOV	Dover AFB, DL	C-5 Base/On-load	Unlimited	5,000	15,000
KCEF	Westover, MA	Enroute	Unlimited	N/A	N/A
KCHS	Charleston AFB, SC	C-141 Base	Unlimited	N/A	N/A
KWRI	McGuire AFB, NJ	Enroute	Unlimited	N/A	N/A
KFOE	Fort Riley, KS	On-load	Unlimited	5,000	15,000
KNON	Notional U.S.	On-load	Unlimited	5,000	15,000
EXXX	Notional Europe	Enroute	500	N/A	N/A
EDAF	Frankfurt, Germany	On-load	144	5,000	15,000
EDAR	Ramstein, AB	On-load	42	5,000	15,000
LETO	Torrejon, AB	Enroute	160	N/A	N/A
LEZA	Zaragoza, AB	Enroute	24	N/A	N/A
OEDR	Dhahran, SA	Off-load	40	2,500	11,000
OEJB	Jubayl, SA	Off-load	15	1,500	7,300
OEDF	King Fahd, SA	Off-load	40	1,400	4,100
OEKK	King Khalid, SA	Off-load	31	450	1,600

The EXXX and KNON airfields are notional for a number of European and U.S. airfields (respectively) with significant involvement in the airlift. The MOG information represents an aggregation of parking and working MOG at each airfield.

Primary Routes. The 32 primary routes used by the airlift aircraft included as many as six stops to cover the 7,000 to 10,000 miles between the U.S. and the Mideast and return. In the GAMS model, a route consists of a sequence of airfields with the first and last airfield being the operating base of the aircraft. The second stop is normally the on-load airfield. A typical ACEP representation of a route is shown in Table 11. The flight time between airfield pairs must be included to compute UTE rates. For the route in Table 11 (route R001 in the model), the round trip mission requires approximately 13.7 hours of flight time during the first 24 hours, 17.2 hours during the second 24 hours, and 6.6 hours on the third day. Similarly, the delay at each stop must be known to compute the time period after the start of thc route when each airfield will be visited. The ground time at each stop is a function of the aircraft type and the purpose of the stop. In the case where more than one type of aircraft can use the same route, the flight times and ground times are averaged rather than including separate routes for each aircraft type. This is done to help reduce the size of the model so that the required response time may be met (Appendix D contains a table of flight times between airfields and ground times).

TABLE 11

Typical Desert Shield Route
(Route #1 – C-5 only)

Route Leg	Start Airfield	End or Activity	Fly Time	Ground Time	Cumulative Time	Day of Visit
1	KDOV	KFOE	3.1		3.10	[0]
	KFOE	On-load		4.25	7.35	
2	KFOE	KCEF	3.1		10.45	
	KCEF	Enroute		3.25	13.70	
3	KCEF	LETO	7.5		21.20	
	LETO	Enroute		3.25	24.45	[1]
4	LETO	OEDR	7.0		31.45	
	OEDR	Off-load		3.25	34.70	
5	OEDR	LETO	8.2		42.90	
	LETO	Enroute		3.25	46.15	
6	LETO	KDOV	8.6		54.75	[2]

Note that each airfield visited on a route is affected by the visit for an entire unit of time. For example, in the Desert Shield model (one time unit = one day), the MOG and throughput (when applicable) available at each airfield visited on each day of the mission is reduced for the entire day. Using smaller units of time reduces the effect but increases the size of the model. This demonstrates another of the trade-offs that must be made between modeling accuracy and model size.

Aircraft Resources. The C-5 Galaxy and C-141 Starlifter represented the primary strategic airlift capability of the U.S. Air Force during the conflict. In addition, aircraft from the civil reserve air fleet (CRAF) were activated for duty throughout the operation. Table 12 lists the primary strategic airlift available and also lists the preferred cargo type and maximum designed capacity, and non-preferred cargo type and capacity where appropriate.

TABLE 12

Desert Shield Aircraft Resources
(Source: AMC/Command Analysis Group)

Aircraft		Preferred		Non-preferred	
Type	Home Airfields	Cargo Type	Capacity	Cargo Type	Capacity
C-5	KDOV,KNON,	Outsize	68.9 tons	Passengers	75
	EDAF	Bulk	68.9 tons	Passengers	75
C-141	KCHS,EDAR	Bulk	27.5 tons	Passengers	10
		Passengers	136		
CRAF 747	KDOV,KNON	Bulk	87.3 tons		
		Passengers	365		
CRAF 707	KDOV,KNON	Bulk	41.1 tons		
CRAF DC-10	EXXX	Passengers	235		

Note that while the home base of the aircraft is known, the Desert Shield schedulers often repositioned aircraft to take advantage of available crews. Consequently, the aircraft availability and UTE constraints are enforced only over aircraft types, and not by operating base. Table 13 lists the total number of each type of aircraft available for Desert Shield missions during the first six months, the MOG used by each aircraft type, and the objective sustained UTE rates.

Movement Requirements. The movement requirements are divided into three classes: outsize cargo, bulk cargo, and passengers. Oversize cargo is included with the bulk cargo because the amount of oversize cargo is not normally known during deliberate planning. The

TABLE 13

Desert Shield Aircraft – Number, MOG and UTE
(Source: AMC/Command Analysis Group)

Aircraft Type	Number Available	MOG Used	Objective UTE Rate
C-5	112	2.0	9.0
C-141	230	1.0	10.0
CRAF 747.	20	2.1	10.0
CRAF 707	15	1.0	10.0
CRAF DC-10	5	2.0	10.0

units from which the requirements originate are notional. Each combination of on-load airfield and off-load airfield is represented by a unit. The data was obtained by working backward from a list of the actual airlift requirements for a typical 30 day period. Table 14 shows the notional units used to represent the valid combinations. The outsize and bulk cargo requirements are expressed in tons. Note that the average monthly capability of the Desert Shield airlift was 61,203 tons of cargo and 71,167 passengers. Thus the total amount of airlift requirements in each category exceeds the capability of the system by some 25,000 tons of cargo and 7600 passengers, as was typical throughout most of the first 180 days.

TABLE 14

Notional Desert Shield Military Units and Requirements
(Source: AMC/XPYR)

Unit Number	On-Load Airfield	Off-Load Airfield	Outsize	Bulk	Passengers
U01	KFOE	OEDR	5,670	6,500	6,850
U02	KFOE	OEJB	3,276	2,800	3,536
U03	KFOE	OEDF	2,646		25,930
U04	KFOE	OEKK		3,375	
U05	KDOV	OEDR		36,378	25,160
U06	KDOV	OEJB		4,582	2,176
U07	KDOV	OEKK		9,704	5,168
U08	EDAF	OEDR	1,764	2,079	7,178
U09	EDAF	OEJB	882	252	1,224
U10	EDAF	OEKK	882	567	1,564
U11	EDAR	OEDR		3,393	
U12	EDAR	OEJB		1,034	
U13	EDAR	OEKK		1,307	
		TOTALS	15,120	71,971	78,786

Analysis of Model Performance

To begin the task of evaluating the performance of the Desert Shield model against "live" data, the model was solved using the primary objective function of minimizing shortfall. The model's performance in solving this problem was evaluated against the established validation criteria in four areas – ease of use, response time, accuracy, and output.

Ease of Use. Information that is required for input to the model must be easily obtainable from sources already available in the planning process and not require extensive transformations prior to input. For the most part, all of the information required to generate a solution to a given airlift scenario is easily available to the airlift planners. The model is flexible in that it can accommodate almost any level of detail. In general, more detailed information on the movement requirements, airfield support capacities, and pickup time windows should result in a more accurate solution, but even very rough and incomplete information can be used to build a model that provides some useful information. Some parameter data for the model may be difficult to develop, however. For example, the parking MOG for each airfield is relatively constant and can be found in a directory of airfields, but working MOG is dependent upon many factors including any simultaneous support the airfield must provide to aircraft not involved in the airlift under study.

To fully utilize the power of the model requires a significant amount of data preparation. Each route must be analyzed in a manner similar to Table 11, where the flight time between airfields and ground times during stops are estimated. In addition, when the flight times and ground times differ significantly by aircraft type, a separate route should be included for each type of aircraft that can fly the route. This can greatly increase the number of routes required. Overall, however, the amount of data preparation required to develop a solution to an airlift of the magnitude of Desert Shield was not unreasonable. The real judges of how easy the model is to use are the airlift planners at AMC.

Response Time. The model must provide a solution to a problem of realistic size within a reasonable amount of time, with a plausible goal being to provide a solution to a 30 day planning problem within 30 minutes of clock time on a minicomputer. The GAMS code executed to obtain a solution to the Desert Shield scenario contained two models – the first to minimize shortfall, and the second to maximize flow using the same data. On average, an optimal solution was found and reported by GAMS in approximately 25 CPU minutes running on a VAX 11/785 minicomputer.

The primary factors in the solution time of any linear program are the size of the model (in terms of the number of variables and the number of constraints), and the density of the constraint matrix. Table 15 shows the model size and density of the two GAMS models. The number of constraints in the model is primarily a function of the number of movement requirements, routes, aircraft types, and time periods. Of these, the airlift planner really only controls the time periods used by deciding on the unit of time. A smaller unit of time will result in more constraints and a larger model. The number of variables is also a function of the movement requirements, routes, aircraft types, and time periods, as well as the cargo classes. However, the number of variables which are candidates to become basic (nonzero) in the solution is controlled by the pickup window of the requirements (DELTA). Thus, tighter and more detailed pickup windows will also constrain the solution, resulting in a reduced solution time.

TABLE 15

Desert Shield – Model Size and Density

Model Objective	Number of Variables	Number of Constraints	Constraint Matrix Density
Minimize Shortfall	3028	2020	0.00579
Maximize Flow	3001	1993	0.00565

Accuracy. The accuracy of an airlift model can be difficult to determine even when compared against historical data for two reasons. First, the model provides an *optimal* solution to a given problem, whereas the historical solution is proven only to be *feasible*. Thus, even if the model is a perfect representation of the real problem, the model solution may be very different from the historical solution. Second, there is likely to be a large number of feasible model solutions with objective function values equal to (alternative optimal solutions) or nearly equal to the one optimal solution reported. Any detailed analysis of the solution must consider these facts.

The primary solution characteristics of interest are the total amount of cargo and passengers delivered (the airlift capability of the system), the size and composition of the fleet of transport aircraft used and (to a lesser extent) the utilization of the aircraft and, finally, the airfields which are bottlenecks in the system. Where the characteristic can be quantified, the goal of the model sponsor is to predict the characteristic to within 10 percent of the "real" value. The first step is to solve the model and compare the model solution to the Desert Shield operation. The model solution to the Desert Shield airlift problem is compared to factual data in Table 16.

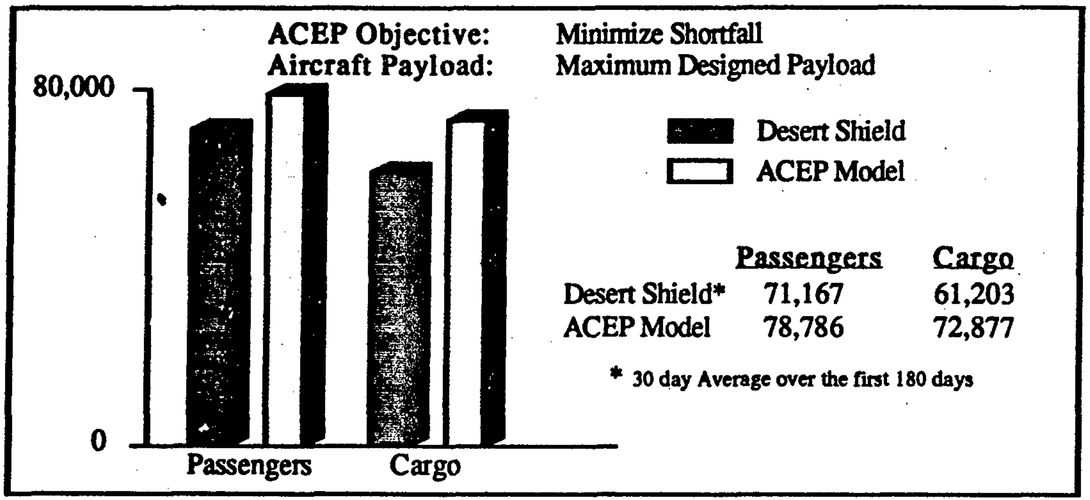
TABLE 16

ACEP Desert Shield Solution Summary #1
(Using Maximum Designed Aircraft Capacities)

Solution Characteristic	ACEP Minimize Cost	Desert Shield
Objective Function Value	143,340.89	337,030 *
Total Number of Missic s	1200	1960

*Estimated by plugging the average D.S. snortfall and number of missions into the Min Cost Obj Function.

Figure 13 shows graphically a comparison of the amount of cargo and passengers delivered. The model has found a solution which delivers 10.7% more passengers, 19.1% more cargo, and uses 38.7% fewer missions than the actual operation. Also, the way in which ACEP uses the aircraft to deliver the cargo is very different from the way it was actually accomplished. Figure 14 shows a comparison of each aircraft's share of the total number of missions flown over the six month period. The information is aggregated into four aircraft categories – C-5, C-141, Wide Body (WB) which represents the CRAF 747 and DC-10 aircraft, and Narrow Body (NB) which



	Passengers	Cargo
Desert Shield*	71,167	61,203
ACEP Model	78,786	72,877

* 30 day Average over the first 180 days

Figure 13. Comparicon of Amount Delivered (Max Aircraft Payload)

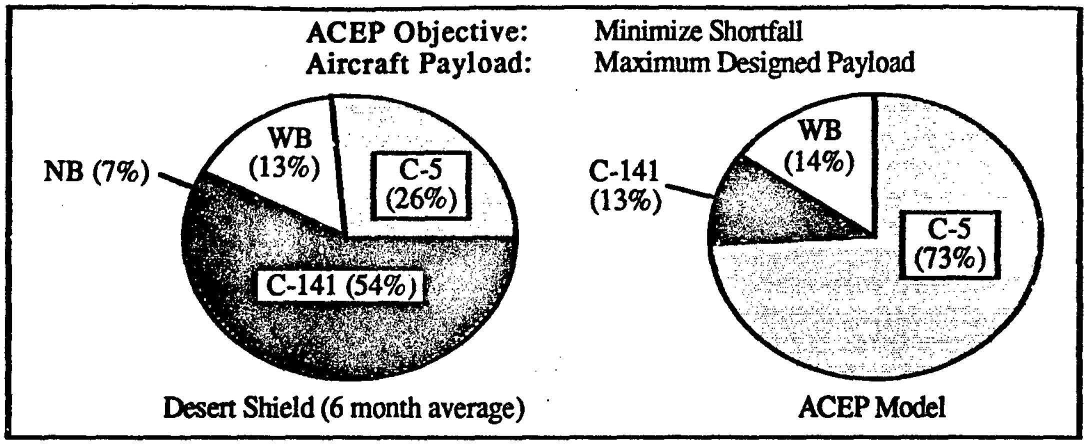


Figure 14. Percent of Total Missions by Aircraft Type

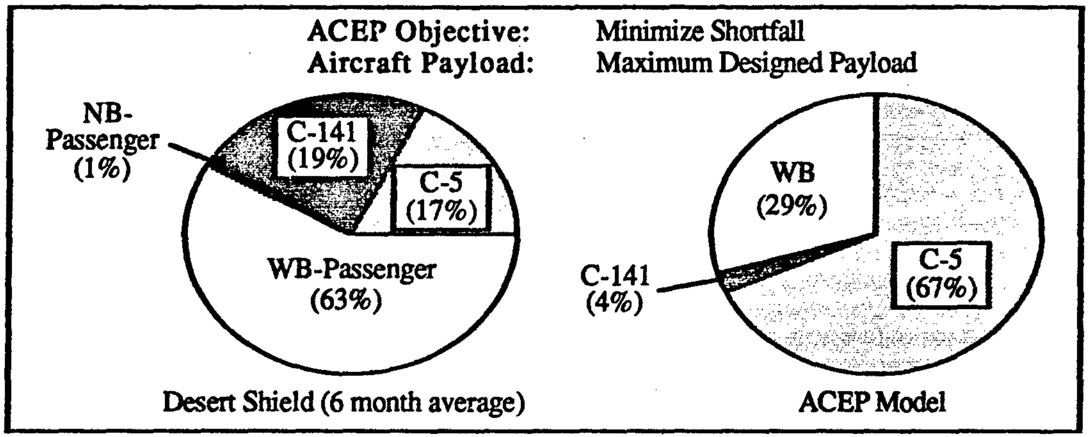


Figure 15. Percent of Total Passengers Carried by Aircraft Type

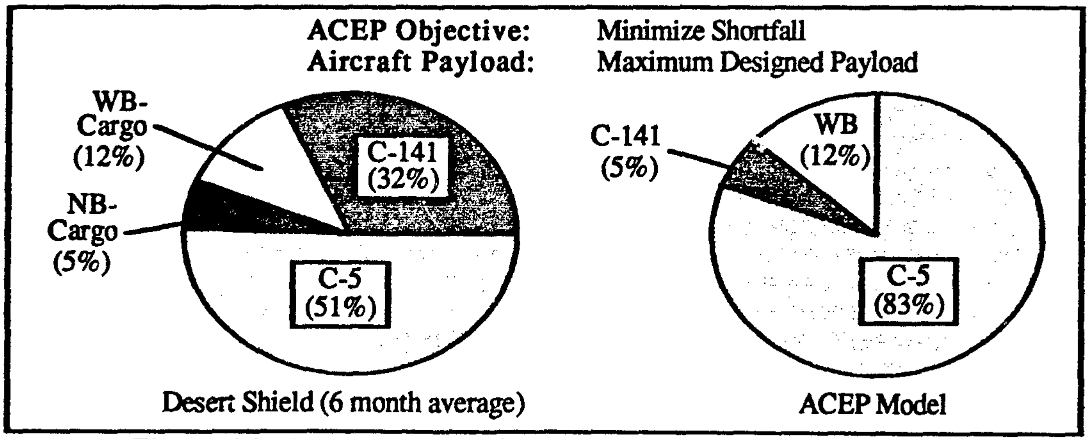
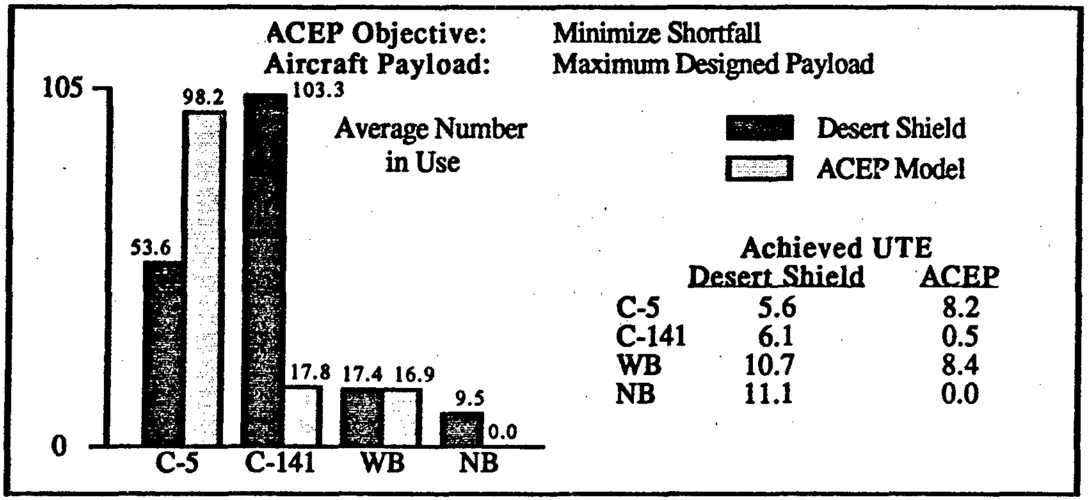


Figure 16. Percent of Total Cargo Carried by Aircraft Type

represents the CRAF 707 aircraft. It is evident from Figure 14 that the model tends to favor the C-5 aircraft and avoids use of the lower-capacity C-141 and narrow body 707 aircraft. Similarly, Figures 15 and 16 show the share of passengers and cargo, respectively, carried by each aircraft type in both the Desert Shield airlift and the model solution. Again the model solution relies heavily on the C-5, and uses none of the available 707 aircraft. Finally, Figure 17 presents a comparison of aircraft utilization. Again, it is clear that the model favors the high-capacity C-5 and wide body CRAF aircraft over the C-141 and narrow body CRAF aircraft.



Achieved UTE

	Desert Shield	ACEP
C-5	5.6	8.2
C-141	6.1	0.5
WB	10.7	8.4
NB	11.1	0.0

Figure 17. Summary of Aircraft Utilization (Max Aircraft Payload)

The results of this initial model beg the question: Why did the airlift planners in Desert Shield schedule nearly 2000 missions each month using an average of 184 aircraft each day when a substantially greater amount of cargo and passengers can be delivered with 40% fewer missions and 50 fewer aircraft? Obviously some important considerations in the scheduling process that determine the size and composition of the fleet necessary to perform a real airlift like Desert Shield are not accounted for in the model.

At least part of the answer lies in the apparent inability of the load planners to fully utilize the payload capabilities of the aircraft. Table 17 compares the maximum designed payload of the strategic airlift force with the average payload achieved in Desert Shield. A new ACEP model was developed and the results of using the actual achieved payload information in the model shows that the model does a remarkable job of predicting the flow capability of the airlift system, but achieves the flow with a significantly different fleet of aircraft.

TABLE 17

Desert Shield Achieved Payloads
(Source: AMC/Command Analysis Group)

Aircraft Type	Cargo Class	Maximum Designed Payload	Desert Shield Achieved Payload
C-5 *	Outsize/Bulk	68.9 tons	62 tons
C-141 **	Bulk	27.5 tons	19 tons
	Passengers	136	112
WB 747	Bulk	87.3 tons	75 tons
	Passengers	365	286
NB 707	Bulk	41.1 tons	24 tons
WB DC-10	Passengers	235	180

* 24 of the 75 passenger seats on the C-5 (non-preferred cargo) were filled on average.

** C-141 aircraft carrying bulk cargo also carried 10 passengers on average.

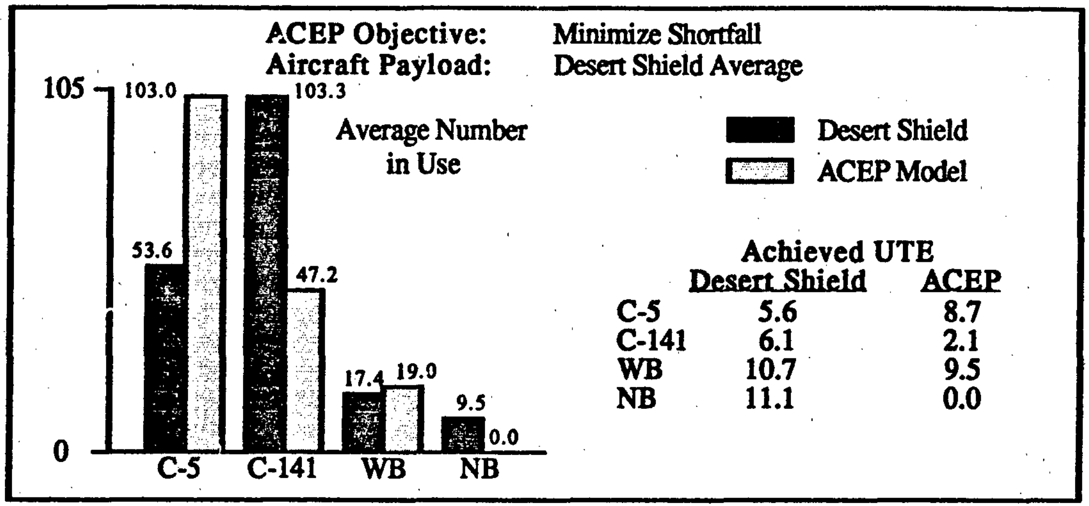
Table 18 compares the ACEP solutions obtained using the maximum designed payloads from Table 12, the Desert Shield achieved payloads from Table 17, and the actual Desert Shield solution which is based on historical records from the airlift.

TABLE 18

ACEP Desert Shield Solution Summary #2

	——Aircraft Payload——		Desert
Solution Characteristic	Maximum	Achieved	Shield
Objective Function Value	143,340.89	318,482.75	337,030
Passengers Delivered	78,786	72,141	71,167
Tons of Cargo Delivered	72,877	62,041	61,203
Total Number of Missions	1200	1530	1960
Average Number of Aircraft	133	169	184

The delivery capability of the new model solution using the achieved aircraft payloads is much closer to Desert Shield figures. The model is able to predict the number of passengers delivered to within 974 passengers (1.7%) and the amount of cargo delivered to within 838 tons (1.4%) of the actual 30-day average. The model's prediction of delivery capability is now well within the specified accuracy tolerance of 10 percent, but to achieve this accuracy the model must be developed using an accurate estimate of the achievable cargo load for each aircraft type. In addition, the fleet of aircraft used in the new model solution is still largely comprised of C-5 and CRAF wide body aircraft. Since the larger aircraft carry greater payloads per mission, fewer



Achieved UTE

	Desert Shield	ACEP
C-5	5.6	8.7
C-141	6.1	2.1
WB	10.7	9.5
NB	11.1	0.0

Figure 18. Summary of Aircraft Utilization (Achieved Payload)

missions are required. This is the major reason why 23% fewer missions are needed in the model solution. Figure 18 summarizes the utilization of aircraft in the new model solution.

There are a number of factors that could explain the remaining differences in the solutions of the model and the Desert Shield planners. Some of the possible factors are identified in the final report to Congress on the Persian Gulf war. These are:

- Nearly 60 percent of the cargo delivered by airlift was oversize and could not be carried by commercial (CRAF) cargo aircraft [Conduct of the Persian Gulf War, 1992: F-38].
- The U.S. provided substantial airlift resources, primarily C-5s, to other Coalition members, limiting the number of C-5s available for carrying U.S. cargo and passengers [Conduct of the Persian Gulf War, 1992: E-9].

Other possible factors which might increase the utilization of C-141 aircraft are identified by Major Killingsworth in "Estimating and Supporting Future Airlift Forces" [Killingsworth, 1991]. He describes a number of operational constraints that limited the deployment even before the number of aircraft available became important. These are:

- The inability of the airlift customers to keep up with the airlift. In the early stages, aircraft and aircrew were positioned at the on-load airfields faster than the users could generate loads. The result was "backlogs of MAC aircraft waiting to be loaded . . . on ramps all over the country" [Killingsworth, 1991: 20]. This rush to load aircraft and get them in the air would favor the lower-capacity C-141 aircraft.

- Lack of an in-theater crew stage base and "burn-out" of the aircrew. "Eventually, the high utilization rates started taking a toll, as aircrews pushed their 30- and 90-day flying hour limitations – a situation that was exacerbated by the lack of an in-theater stage crew operation (a base in Saudia Arabia where fresh crews would be available for flying the return leg of the mission)" [Killingsworth, 1991: 20]. While there is no direct evidence that aircrew factors limited the use of the C-5, it is likely that some effort was made to spread the flying hours evenly among the available crews to avoid "burn-out". Again, this would tend to favor the more numerous C-141 aircraft over the C-5.

Using this new information, a new ACEP model was developed and solved which limited the utilization of the C-5 aircraft to the average 6-month UTE rate achieved in Desert Shield. The results are presented in Table 19.

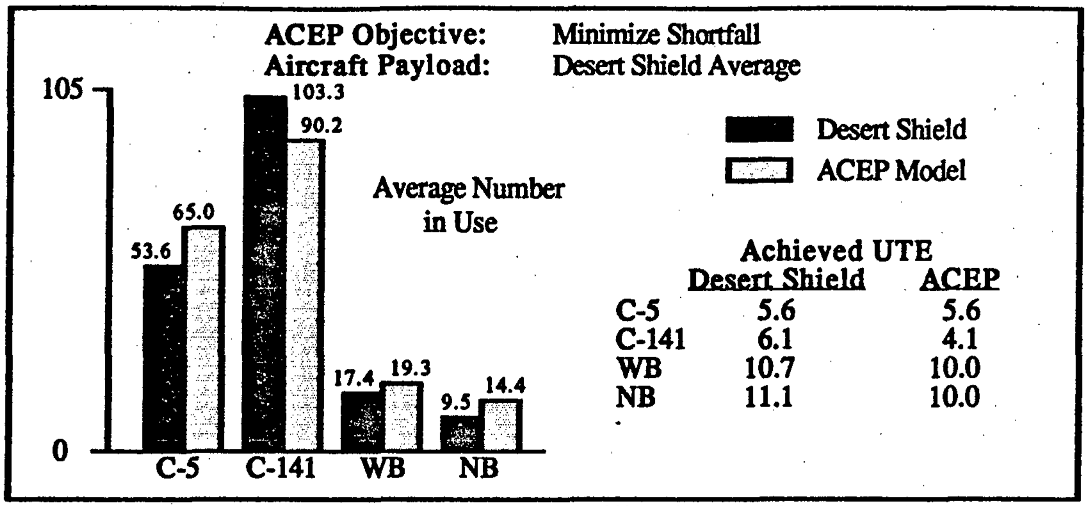
TABLE 19

ACEP Desert Shield Solution Summary #3

	— Aircraft	Payload —	Limited	Desert
Solution Characteristic	Maximum	Achieved	C-5 UTE	Shield
Objective Function Value	143,340.89	318,482.75	325,852.84	337,030
Passengers Delivered	78,786	72,141	71,761	71,167
Tons of Cargo Delivered	72,877	62,041	61,715	61,203
Total Number of Missions	1200	1530	1840	1960
Average Number of Aircraft	133	169	189	184

The model solution is now roughly equal to the solution used in Desert Shield. Figures 19 and 20 show that the aircraft utilization and the contribution of each type of aircraft to the airlift are approximately equal. However, to achieve this solution the C-5 aircraft UTE rate must be constrained below the maximum sustained rate.

The new solution provides a valuable discovery – the Desert Shield solution is a feasible solution in the model! Had it not been so, the validity of the model would be much harder to justify. However, to achieve the Desert Shield solution using the model, factors such as loading delays and aircrew "burn-out" have been incorporated indirectly by adjusting the C-5 UTE rate. It is unclear to what extent airlift planners would have advance knowledge of such factors.



Achieved UTE

	Desert Shield	ACEP
C-5	5.6	5.6
C-141	6.1	4.1
WB	10.7	10.0
NB	11.1	10.0

Figure 19. Summary of Aircraft Utilization (C-5 UTE = 5.6)

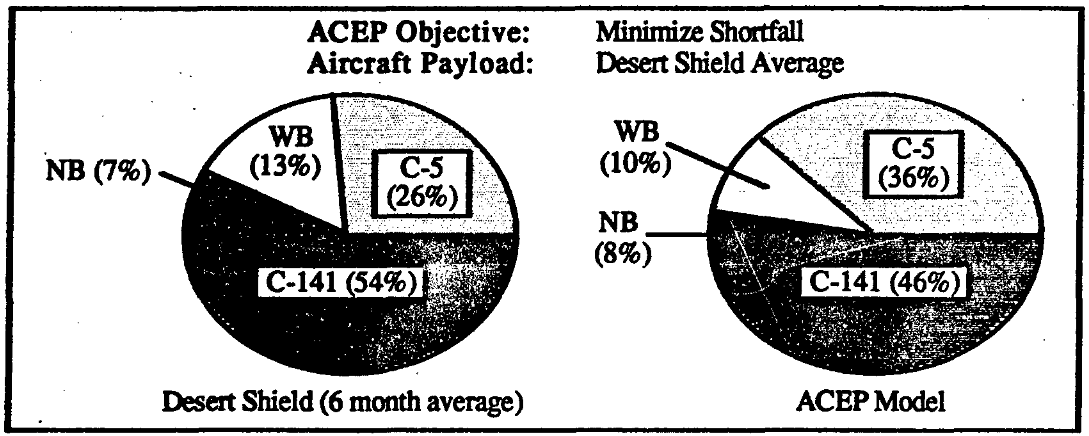


Figure 20. Percent of Total Missions by Aircraft Type (C-5 UTE = 5.6)

The model was more consistent in identifying the airfield bottlenecks in the system. Each model solution was approximately the same in this regard. The bottlenecks identified by the model in each of the throughput constraint categories – cargo and passenger throughput – are presented in Figures 21 and 22. Note that only King Khalid airfield (OEKK) reached a limit, but the marginal value of the throughput constraint at this airfield indicates that a significant improvement in objective function could be experienced by increasing the cargo handling capability (MHE) of the airfield. The final report to Congress on Desert Shield states that shortages in some types of MHE at the off-load airfields (specific airfields not identified) resulted in extended ground times early in the airlift [Conduct of the Persian Gulf War, 1992: F-25].

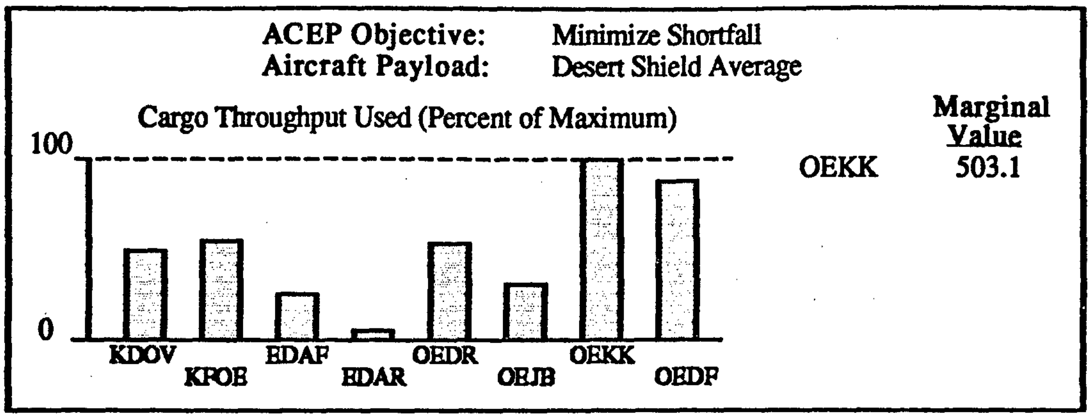


Figure 21. Summary of Airfield Cargo Throughput

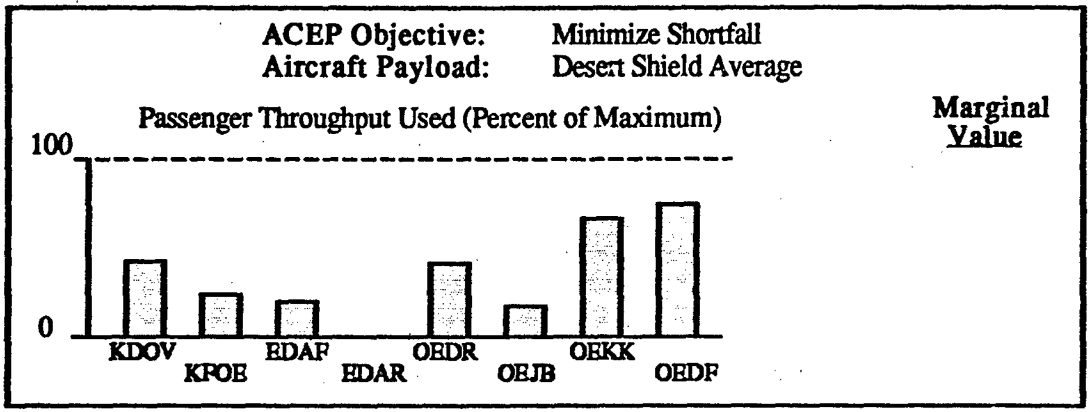


Figure 22. Summary of Airfield Passenger Throughput

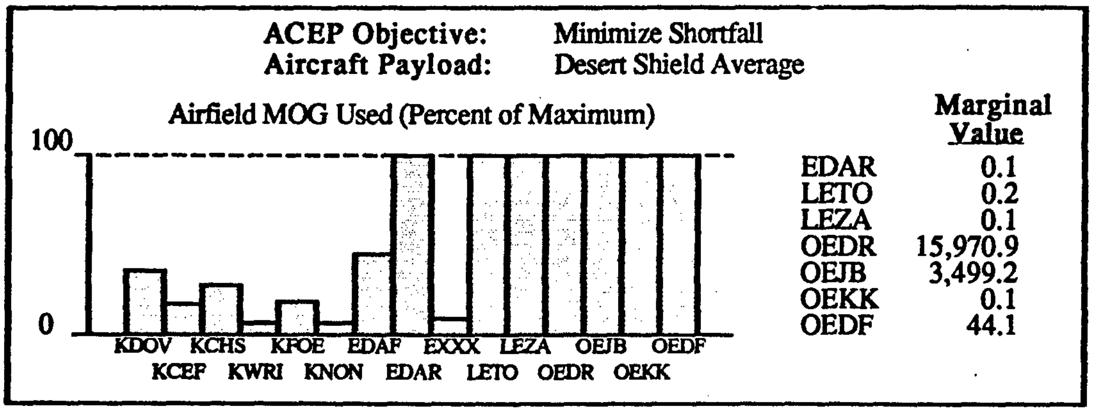


Figure 23. Summary of Airfield MOG

Many of the enroute airfields and all of the off-load airfields in Saudi Arabia reached their MOG limits at some time during the planning period of the model. Figure 23 provides a summary of the utilization of MOG (parking and working MOG) at each airfield and the aggregated marginal value for each airfield that reached 100 percent of MOG capacity. The airfields are listed with the on-load airfields on the left, the enroute airfields in the middle, and the off-load airfields on the right. The graph makes it very obvious that the enroute and off-load airfields represent the primary bottlenecks in the airlift system. The marginal values indicate that the most critical of the MOG-constrained airfields are the off-load fields in Saudi Arabia. In fact, the marginal value for Dhahran (OEDR) indicates that the objective function can be improved by as much as 15,970 (5%) simply by increasing the MOG at this airfield by one unit across the 30 days (again this assumes no change in the optimal basis – the actual improvement is 10,664.0). The model's identification of these airfields as the most critical bottlenecks in the airlift matches well the situation found in Desert Shield [Conduct of the Persian Gulf War, 1992: E-8].

To summarize the findings on the accuracy of the ACEP model, the model seems to perform well with the limited information that would be available to airlift planners during deliberate planning only in predicting the flow capability and the airfield bottlenecks. Even in the deliberate planning mode, an estimate of the operational payload capability of the aircraft must be known. When additional information is available about loading delays, aircrew utilization, oversize cargo requirements, and other factors limiting the availability of aircraft to perform airlift missions such as support to allied forces, then the model is also able to predict the size, composition of the airlift fleet needed and the expected utilization of the fleet.

Output. The GAMS modeling language is a powerful tool in manipulating and presenting the model solution. GAMS builds a database of information on the model solution that includes records on each variable and equation [Brooke et al., 1988: 21]. There are four fields within each record which are updated by GAMS when a solution is returned from the solver. These fields are:

(1) lower bound on variable or constraint right-hand-side;

(2) the current (or optimal) level of the variable or constraint;

(3) upper bound on variable or constraint right-hand-side; and

(4) the marginal or dual value of the variable or constraint.

GAMS allows the modeler complete read- and write-access to the database, making the transformation and display of the solution easier. For example, the following information is computed on the ACEP Desert Shield model solution by manipulating the database (see Appendix F, ACEP Desert Shield Solution):

- PAVAIL – the maximum number of each type of aircraft used at any time during the model planning period.
- AAVAIL – the average number of each type of aircraft used during each UTE period.
- MAVAIL – the marginal value of each aircraft type summed across the planning period.
- PCTHRU/MCTHRU – the maximum cargo throughput used (percent of available) and the aggregated marginal value at each on-load and off-load airfield.
- PPTHRU/MPTHRU – the maximum passenger throughput used (percent of available) and the aggregated marginal value at each on-load and off-load airfield.
- PMOG/MMOG – the maximum MOG used (percent of available) and the aggregated marginal value at each airfield.
- TOTAMT – total amount of cargo and passengers waiting for airlift.
- UTOTAL – total shortfall in each cargo class.
- DTOTAL – total amount of cargo and passengers delivered over the planning period.
- SORTIES – number of each aircraft type scheduled to begin an airlift mission on each day in the planning period (schedule of airlift missions).
- TOTALS – number of airlift missions performed by each aircraft type.
- TSORT – total number of airlift missions performed.
- MIXTURE – percent of total number of missions flown be each aircraft type.
- PERCENT – percent of total amount in each cargo class delivered by each aircraft type.
- ACTUTE – actual UTE rate achieved by each aircraft type in the solution.
- FLYRATE – average number of flight hours accumulated per day by each aircraft type.

This information is very useful in analyzing the airlift problem and the ACEP solution. For example, the "fly rate" can provide valuable information about how well the airlift problem has been modeled. The fly rate for an aircraft can be defined as the average daily flight time accumulated by the aircraft *actively* participating in the operation. Fly rate is primarily a function of the routes and the cycle time. Comparing the ACEP fly rate to the actual average flight time for each aircraft type in Desert Shield reveals how well these aspects of the problem have been modeled. Table 20 provides a comparison of the fly rates in the third (and most accurate) ACEP model to the fly rates achieved in Desert Shield. In this case, the fly rates of the model solution are consistently low, indicating that the cycle times used on the routes are too high. However, a non-integer cycle time must be used to model the fly rates accurately, and a non-integer value for this parameter is not allowed in the GAMS model implementation. One way to improve the accuracy is to use smaller units of time, but this increases the size of the model dramatically.

TABLE 20

Comparison of Fly Rates

Aircraft Type	ACEP Model	Desert Shield
C-5	9.6	10.4
C-141	10.2	11.9
WB	10.1	10.7
NB	10.3	11.1

In conclusion, the retrospective test applied to the ACEP model provided valuable insight into the expected performance of the model in a sustained airlift situation. The model is relatively easy to build and solve, and solutions to the 30-day problem examined were computed relatively quickly. In addition, the GAMS modeling language provides a powerful medium for examining the model solution. On the other hand, the accuracy criteria specified by the model sponsor was largely not met with only deliberate planning information. When the operational payload of the aircraft can be estimated, then the model may provide an accurate (within 10 percent) estimate of the flow capability of the airlift system and help identify and rank order the bottlenecks in the system. When additional information is available, such as loading delays and aircrew availability, then the model may also be able to provide an estimate of the fleet size and composition. Specific recommendations on the use of the ACEP model for airlift planning are made in the next chapter.

VI. *Recommendations and Conclusion*

The ACEP model has now been through one full iteration of the proposed model development life cycle. Many iterations may be required to validate the model to the sponsor's satisfaction, and new iterations should be performed to maintain the model. However, if the validation process has been successful, then the model will have already reached a level of maturity such that it can make a significant :ontribution to the airlift planning problem. To conclude the research, the validation paradigm used on the ACEP model is reviewed followed by a summ.ary of the validation findings. Specific recommendations are made on the use of the ACEP model for airlift planning. Finally, recommendations for further research into the validation problem as well as the airlift planning problem are given.

Review of the Validation Paradigm

The focus of this research included two main issues. The first of the two major problems was to propose a methodology for evaluating the validity of an existing model. The methodology proposed includes a new model development life cycle and techniques for applying the life cycle to existing models.

Overview of New Life Cycle. The model development life cycle proposed as part of the validation process includes four stages which are designed to provide a controlled transformation of the model from its conceptual form to an operational form that solves the "right" problem. The full life cycle is shown once again in Figure 24. While maintenance of the model could be included as a fifth stage, it is more proper to consider the maintenance process as a microcosm of the full life cycle. Each of the stages is summarized as follows:

(1) Requirements Analysis. The primary goal of this stage is to analyze and document the problem or situation being modeled and the model sponsor's requirements for solving the problem. The validation process is started here by establishing specific criteria for the performance of the model and identifying the post-development tests that will best determine the validity of the model.

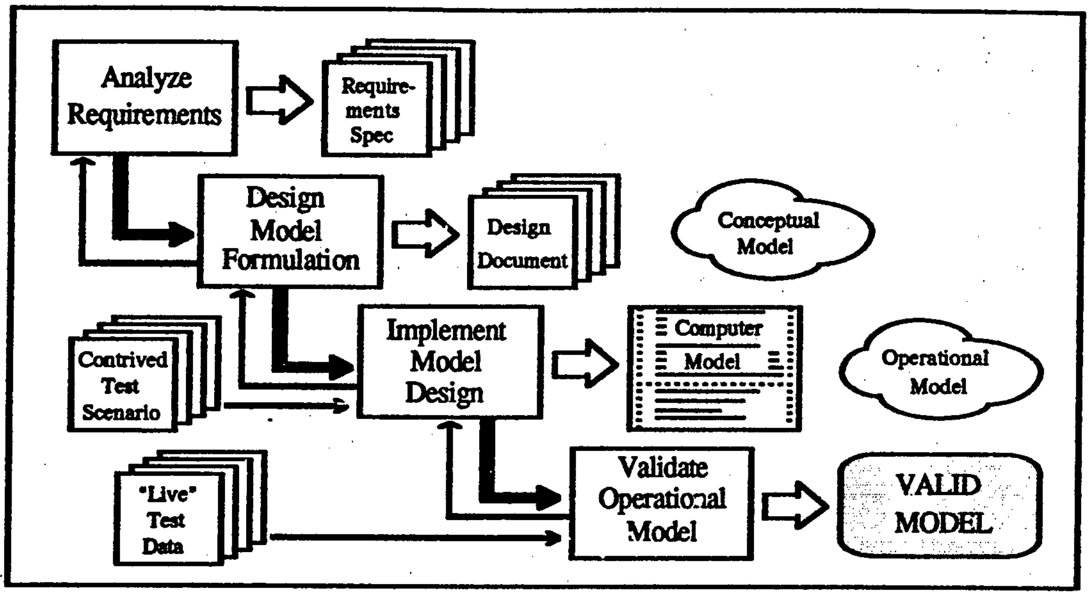


Figure 24. Model Life Cycle Paradigm

(2) Model Design (formulation). The outcome of this stage of the process is a conceptual model. Again, the model design must be documented to p~~r~~ovide the necessary continuity in the life cycle process. The conceptual model details the modeling approach taken and the assumptions required to use the approach.

(3) Model Implementation. After the conceptual model has been verified to meet the needs of the model sponsor, it must be implemented in an operational (i.e. executable) form. The main validation goal of this stage is to ensure that the operational model is a valid representation of the conceptual model.

(4) Operational Validation. Finally, the performance of the model on a "live" situation or set of data is compared to the criteria established in the requirements analysis stage. If all criteria are meet, then the model is ready for use. Otherwise, the life cycle is backtracked to the point where corrections can be made to meet the criteria.

This four-stage life cycle offers significant advantages over the traditional model development approach discussed in Chapter II. First, the model formulation is divided into two distinct phases – analysis of requirements and model design. This places more emphasis on the investigation of requirements in the beginning, increasing the chances that the operational model will be "useful" to the model sponsor. Second, the results of each stage of development are documented. This allows for a "face validation" of the mod · · earlier and more often. In addition, the transformation from conceptual model to operational model is more controlled. Finally, the documentation makes maintenance and use of the mod^el easier and an independent evaluation of the model possible. The proposed life cycle can be applied to existing models as well.

Life Cycle Recovery Process. In Chapter III, techniques were presented for applying the life cycle to models under development or already developed. These techniques were used to recover the missing life cycle documentation so that the advantages of the new, four-stage life cycle could be realized. Three approaches were presented for reconstructing the missing life cycle documentation:

(1) Starting the life cycle over beginning with the analysis of requirements and ignoring any previous work or existing models.

(2) "Backing in" to the missing documentation based upon the existing model and making changes identified through interaction with the model's sponsor.

(3) A combination of both approaches.

The primary lesson of the research is that validation of models is a difficult and tedious process and cannot be totally separated from the development life cycle of the model. Thus, the only hope in conducting an independent evaluation of a model is to have (or recreate) documented model requirements approved by the model sponsor, a written formulation that explains how the requirements are met in conceptual form, an operational model verified to correctly implement the formulation, and the results of any previous post-development validation tests performed.

Summary of ACEP Validation Findings

The ACEP model proposed by Oak Ridge National Laboratory provided a good case study of the life cycle recovery process. The results of each stage of the methodology used to validate the model are summarized as follows:

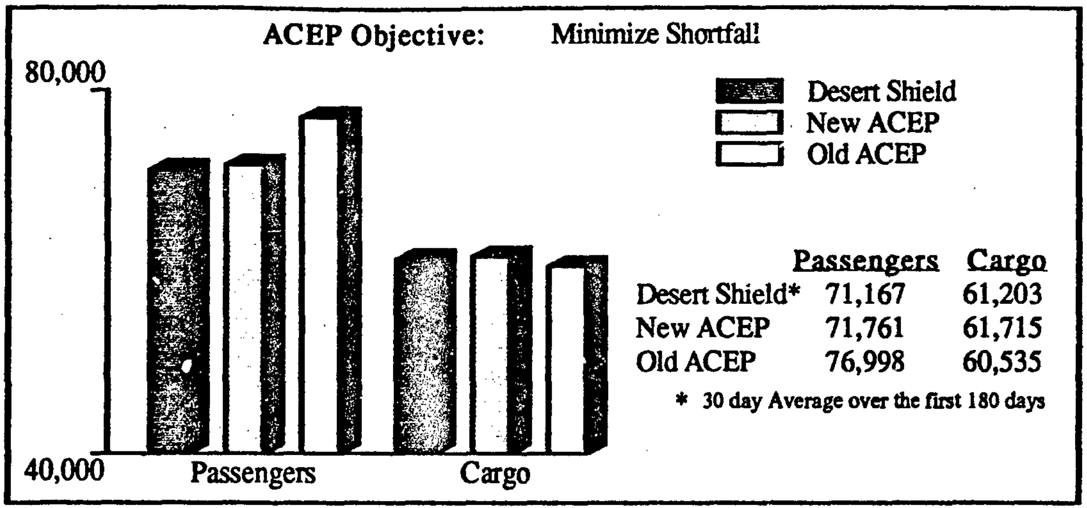
(1) Recovery of Model Requirements. Model requirements were developed by backing into a requirements document and making the changes requested by the model sponsor at AMC. Additional constraints to account for airfield working MOG and aircraft UTE

were added to the model as a result of the requirements analysis process. In addition, specific validation criteria were established.

(2) Model Implementation. The GAMS operational model was validated using three techniques collectively called "constraint validation". These include the development of a contrived scenario, the examination of the effect of individual constraints, and the use of alternative objective functions. Improvements made to the ACEP model as a result of the constraint validation process include modifications to the objective function to minimize closure, additional constraints to enforce aircraft UTE and availability by operating base as well as across aircraft types, an additional constraint to offset the model's natural tendency to favor the delivery of passengers over cargo, and improved presentation of the optimal soloution.

(3) Post-Validation Test. The retrospective test used "live" data from the Operation Desert Shield airlift to provide further insight into the validity of the model. During the test, the model solution was compared to the Desert Shield airlift in terms of the model objective function, the amount of cargo and number of passengers delivered over 30 days, the number of airlift missions performed, the average number of aircraft used, and the contribution of each type of aircraft to the total result. It was discovered that the model can estimate the flow capability of a Desert Shield-type operation within the required tolerance as long as the model is built using an estimate of the "operational" payloads of the aircraft. In addition, the identification of airfield bottlenecks seemed to be accurate. On the other hand, the optimal aircraft fleet size and composition found by the model differed significantly from the fleet used in Desert Shield. A number of factors were found which could explain the difference – such as the high percentage of oversize cargo which could not be carried by CRAF aircraft. Some of the factors which were used to explain the difference required information that would be available only during execution planning of an airlift.

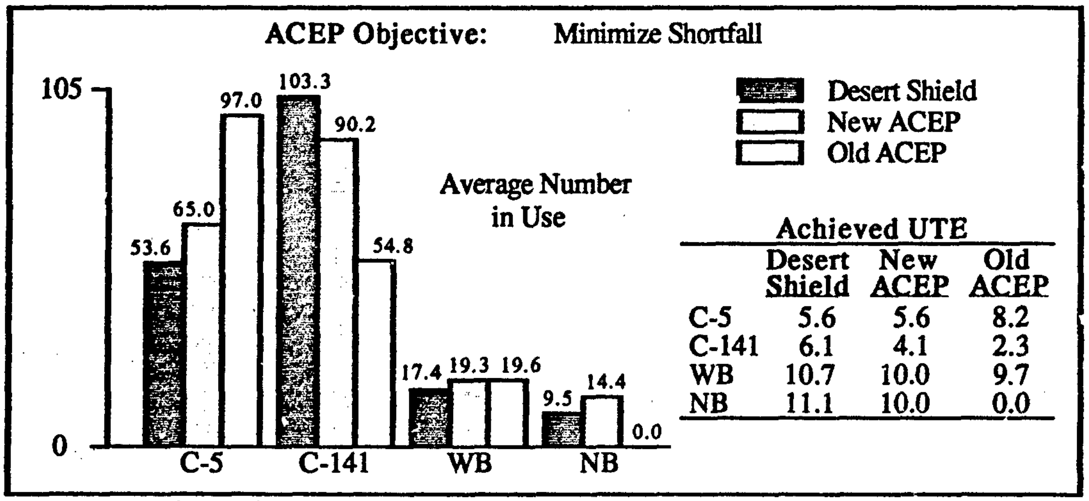
As a result of the validation findings, the model has been improved. While the Desert Shield scenario is not ideal for demonstrating all of the improvements made, the new ACEP model solution is closer to the actual airlift than the original model solution, as shown in Figures 25 and



	Passengers	Cargo
Desert Shield*	71,167	61,203
New ACEP	71,761	61,715
Old ACEP	76,998	60,535

* 30 day Average over the first 180 days

Figure 25. Comparison of Amount Delivered (All Models)



Achieved UTE

	Desert Shield	New ACEP	Old ACEP
C-5	5.6	5.6	8.2
C-141	6.1	4.1	2.3
WB	10.7	10.0	9.7
NB	11.1	10.0	0.0

Figure 26. Summary of Aircraft Utilization (All Models)

26. In addition, the model output has been improved to provide much more concise and useful information about the model solution, making analysis of the solution easier.

Recommendations for Use of the ACEP Model

The ACEP model was tested against one of the largest, most difficult airlift operations in history. However, massive, sustained airlift operations like Desert Shield have occurred only once every 40 years or so. Consequently, the model should be tested against more mundane, day-to-day airlift scenarios before receiving full accreditation for use on the airlift problem.

In addition, the two modes of airlift planning – deliberate and execution – have different and, in many ways, incompatible requirements. Since the ACEP model was developed with both modes in mind, some compromises had to be made. Based on this research, the recommended uses of the ACEP model as it currently stands are:

(1) Deliberate planning. The model should provide reliable results in identifying the most critical airfield bottlenecks in a given airlift system. In addition, the model may be appropriate for conducting *flow capability studies* or *transportation feasibility studies* where the size and composition of the aircraft fleet to be used are known or estimated. Whenever possible, and particularly when CRAF aircraft are involved, the *oversize* cargo category should be included in the model to increase accuracy.

(2) Execution planning. The model seems to have the greatest unrealized potential in this mode of planning. Given that the operational payloads of the aircraft can be estimated accurately and that aircrew utilization factors can be accounted for, then the model may be used to determine fleet size and composition requirements. The primary limiting factor is the effort required to find an optimal solution as the problem size increases.

Recommendations for Further Research

The life cycle paradigm of modeling is not well developed and requires further research to reach the mature paradigm available in other engineering fields – such as software engineering. Some specific recommendations for further research into the life cycle process are:

(1) Develop alternatives to the "waterfall" life cycle. In software engineering several alternative life cycles have been identified that build upon the basic "waterfall" life cycle and may be more appropriate under certain circumstances. Examples include:

- the "rapid prototyping" life cycle which is used when the requirements for the software are not well understood. This life cycle focuses the initial effort on developing a prototype of the envisioned system for the purpose of investigating further the user's true needs.
- the "evolutionary" life cycle where the software is developed in increments or phases.

These alternative life cycle approaches seem particularly applicable to modeling and may be more appropriate than the classic "waterfall" approach.

(2) Documentation standards and configuration management. As mentioned in Chapter II these topics have received a great deal of attention in the software engineering field, but almost none in computer modeling. Some specific questions that might be addressed include:

- Which models need to go through the rigorous life cycle process? If the model's intended use is very limited, then certainly the need for a rigorous, documented process could be questioned.
- What should be documented in each stage of the life cycle and how are the documents updated when a change is made to the model?
- How much should the model sponsor be involved in each stage of the life cycle?

The ACEP validation effort has also yielded some areas where further research on the modeling and solution approaches seems warranted. These are:

(1) In Chapter V it was discovered that non-integer cycle times on the routes may result in a more realistic modeling of the delays associated with post-mission aircraft maintenance and crew rest. The model should be modified to allow airlift planners to use non-integer cycle times.

(2) AMC should consider splitting the model into two models, each developed further to meet the needs of the two types of airlift planning – deliberate and execution. The current ACEP model was developed for both situations and is a compromise of the needs of the two modes of operation.

(3) The model could be modified to include the "direct delivery" concept that has been identified by many as the primary advantage of the new C-17 aircraft (see Cooke, 1984; Miller, 1988; Streater, 1988; Ulsamer, 1984). This model must incorporate the Required Delivery Date (RDD) and account for the travel time between the off-load airfield and the final destination of the passengers and cargo (Appendix A, Section 5.2).

(4) As mentioned in Chapter II, other heuristic techniques for finding good, integer solutions should be investigated. Some techniques from the literature which seem

especially promising are *simulated annealing* and *Lagrangian relaxation*. These techniques offer efficient solution algorithms for finding integer solutions to large problems in a reasonable amount of time.

Appendix A: Model Requirements Specification for the Airlift Capabilities Estimation Prototype (ACEP)

1.0 OVERVIEW

The Airlift Capabilities Estimation Prototype (ACEP) is a deterministic model that provides rough estimates of the resources needed to perform an airlift operation. The ACEP model is used in two modes: (1) during the early stages of deliberate planning when only approximate information on the mobility requirements is available and the response time of the model is not critical; and (2) during execution planning to obtain quick estimates of resources after a modification of mobility requirements. In the second mode, the response time of the model is critical. The output of the model is used for further planning or as the initial estimate of resources required to start more detailed analysis of the problem.

2.0 APPLICABLE DOCUMENTS

2.1 AFR 28-3, USAF Operations Planning Process.

2.2 Busch, Ingrid K. and Michael R. Hilliard. *An Airlift Capabilities Estimation Model for the Military Airlift Problem*. Operations Research Group, Center for Transportation Analysis, Oak Ridge National Laboratory, 1992.

2.3 Gearing, Capt Rick, and Maj Jim Hill. "UTE Rates Revisited," *Airlift:* 18 - 21 (Spring 1988).

2.4 Hilliard, Michael R., Rajendra S. Solanki, Cheng Liu, Ingrid K. Busch, Glen Harrison, and Ronald D. Kraemer. "Scheduling the Operation Desert Storm Airlift: An Advanced Automation Scheduling Support System," *Interfaces, 22:* 131 - 146 (1992).

2.5 MAC Regulation 55-28.

2.6 Smith, Maj Ronny C. "Station Capability," *Airlift:* 14 - 16 (Winter 1985).

3.0 PROBLEM SUMMARY

The airlift planning problem can be described as follows: it is desired to transport a known amount of cargo and passengers from a set of on-load airfields, through a sequence of enroute airfields, to one or more off-load airfields within a specified amount of time. A fleet of heterogeneous aircraft is assigned to perform the airlift, and a set of possible routes to move the cargo from origins to destinations is available. There are many conflicting objectives and constraints in the problem and many formulations are possible. In addition, problems of practical size and importance involve dozens of airfields, hundreds of movement requirements and aircraft, and span over a planning horizon of 90 days or more, making the resulting problem difficult to solve.

4.0 MODELING OBJECTIVE

The primary objective of the ACEP model is to obtain reasonable estimates of the number and type of aircraft needed to perform a specific airlift operation. In addition, the model should be able to identify bottlenecks in the airlift system that are the limiting factors in achieving an optimal flow of cargo and passengers through the system.

4.1 PRIMARY OPTIMIZATION GOALS.

The primary optimization goal of the ACEP model is to minimize shortfall – the amount of cargo and passengers left undelivered at the end of a specific planning period.

4.2 SECONDARY OPTIMIZATION GOALS.

Of secondary concern to minimizing undelivered cargo is the efficiency of the operation. The model should attempt to develop an airlift schedule which minimizes cost and maximizes resource utilization. Other desirable goals are:

- Deliver each movement requirement as early as possible;
- Minimize the number of aircraft used; and
- Maximize aircraft utilization (within limits).

5.0 MODELING CONSTRAINTS

There are both spacial and temporal constraints that effect the number of aircraft required and the scheduling strategy used to employ the aircraft. A discussion of these constraints follows.

5.1 PLANNING HORIZON

Normally, an airlift operation spans a given planning period or horizon. At the end of the period all movement requirements should be delivered to the appropriate destination. Any undelivered cargo or passengers at the end of the period is shortfall.

5.2 MOVEMENT REQUIREMENTS

The cargo to be moved can be classified into four categories: outsize, oversize, and bulk cargo, and passengers. Outsize cargo is equipment too large and heavy to be carried by any aircraft in the Air Force inventory except the C-5 (and the C-17 when it becomes available). An example of outsize cargo is the Bradley armored personnel carrier. Oversize cargo includes equipment such as a 2.5-ton truck that is too large for standard-size pallets. Bulk cargo is everything else which can be consolidated onto standard-size pallets. The deployment of military units normally requires the movement of a mixture of cargo types and passengers.

Cargo and passengers from the various units being moved can only be picked-up and delivered during certain time windows (Figure 27). The earliest time that a payload may be loaded onto an aircraft at the on-load airfield is called the *available-to-load date* (ALD). This date is determined by the military unit to which the cargo or passengers are assigned. The earliest time that the payload can arrive at the off-load airfield is called the *earliest arrival date* (EAD). The *latest arrival date* (LAD) is the latest time that the payload can arrive at the off-load airfield. These times are determined by the commander of the operation. Finally, each requirement may have a *required delivery date* (RDD) which is the deadline for arrival of the requirement at its final destination within the theater of operation [Hilliard, et al., 1992: 133].

Past airlift operations in support of contingencies have shown that approximately one ton of cargo must be delivered for each passenger. This is not true of all airlift scenarios.

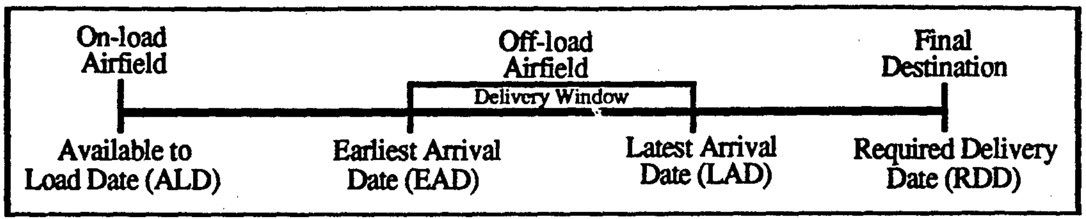

On-load Airfield
Off-load Airfield
Final Destination
Delivery Window
Available to Load Date (ALD)
Earliest Arrival Date (EAD)
Latest Arrival Date (LAD)
Required Delivery Date (RDD)

Figure 27. Airlift Planning Temporal Constraints

5.3 AVAILABLE ROUTES

Many possible routes may be available for the aircraft to move the requirements from the on-load airfields to the off-load airfields. However, the number of practical routes available is usually limited and these routes can be enumerated. This makes the problem much easier to solve from a modeling perspective. An ideal route will start at the aircraft's home base (Figure 28). The aircraft will on-load the cargo or passengers at the airfield nearest to the origin of the cargo or passengers and fly as direct as possible to the off-load airfield. A stop at one or more enroute airfields may be required to refuel or change crews. The off-load airfield is located as close as possible to the final destination of the cargo or passengers. A recovery base is often used for refueling and maintenance purposes to ease the congestion at the off-load airfields. Then the aircraft returns to its home base to prepare for the next mission.

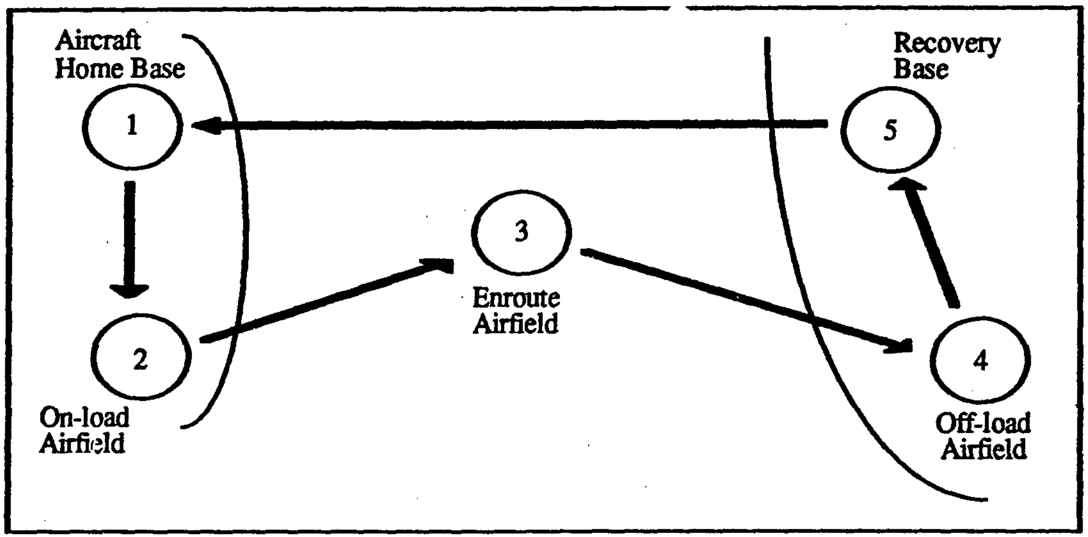


Figure 28. Typical Airlift Route

5.4 ENROUTE SUPPORT AND STATION CAPABILITY

The on-load, off-load and enroute airfields have varying capability to support visiting aircraft. The on-load and off-load airfields are limited in the amount of cargo and passengers that can be processed through the field each day (called throughput). The throughput limit is dependent upon factors such as the capacity of the passenger terminal and the number and type of material handling equipment (MHE) available. The MHE is required to load and unload cargo from the aircraft.

All of the airfields involved in the airlift flow are limited in the number of aircraft that can be supported at a given time. The myriad of factors involved in determining this limit are often aggregated into a single constraint called maximum on the ground (MOG). MOG is defined in MAC Regulation 55-28 as "the highest number of aircraft being used in an operation which will be allowed on the ground during a given span of time based on simultaneous support". Analysts refer to two different types of MOG in planning airlifts. The ramp space available to park aircraft is called "parking MOG". All other aircraft servicing constraints are included in the "working MOG". A number of diverse factors contribute to the working MOG at a particular airfield and point in time. Typically the most important factor is the refueling capability of the airfield. The most constraining of the two types of MOG – parking and working – is the limiting factor in supporting an airlift operation. MOG for a particular airlift operation at an airfield is also effected by other enroute traffic the airfield must support simultaneously [Smith, 1985].

5.5 AIRCRAFT CONSTRAINTS

Size, capacity and maneuverability contribute to the amount of MOG used by a particular type of aircraft. Other airlift planning factors associated with the aircraft used are [Hilliard et al., 1992: 132 - 137]:

(1) Each type of aircraft has a limited capacity for each type of cargo, and some aircraft may be unable to carry some types of cargo.
(2) The flow planners may specify a minimum load required to justify a mission.
(3) Each type of aircraft has a cargo type for which it is best suited to carry (called its *preferred* cargo type).
(4) Certain aircraft may have special capabilities which can be used to improve the airlift flow. For example, C-5 aircraft are capable of carrying as many as 75 passengers in addition to any cargo carried (this is called a *non-preferred* cargo type).

Finally, the aircraft must not be over-utilized. Air Force planners aggregate all the factors that limit the utilization of aircraft into a number called the UTE rate. UTE rate can be defined as the total pool of daily flying hour capability for a fleet of (homogeneous) aircraft distributed equally by each primary aircraft authorization [Gearing and Hill, 1988: 187]. Three types of UTE rates are commonly used: peacetime, objective wartime, and obtainable wartime. The peacetime UTE rate is based solely upon the flying hours approved by Congress in the annual budget. The objective wartime UTE rate corresponds to the capability that planners would like to achieve in wartime. Obtainable wartime rates are the actual airlift capabilities experts believe can be achieved for a particular wartime scenario and set of resources. Objective and obtainable UTE rates are expressed for both surge (first 45 days) and sustained (after 45 days) periods.

Four factors are used to determine the obtainable wartime UTE rates for various aircraft: aircrew manning, maintenance manning, spare parts, and the airlift system. The UTE rate that is most constraining is used as the estimate for deployment capability and included in the Joint Strategic Capabilities Plan (JSCP). The JSCP reports four UTE rates for a given aircraft type and scenario. A UTE rate is calculated for three divisions within the surge period (the first 45 days of an operation) beginning with C-day (the day deployment begins) and an overall surge rate is calculated as follows (Figure 29):

(1) C-day through C-day + 2. UTE rate starts low and builds quickly to a high value reflecting the generation of aircraft into the airlift operation.
(2) C-day + 3 through C-day + 15. Aircraft are assumed to surge to the objective UTE rate due to the critical nature of the early deployment period.
(3) C + 16 through C + 45. A pool of flying hours is calculated which is the total hours an aircraft can fly in the 45 days of surge minus the hours already flown.

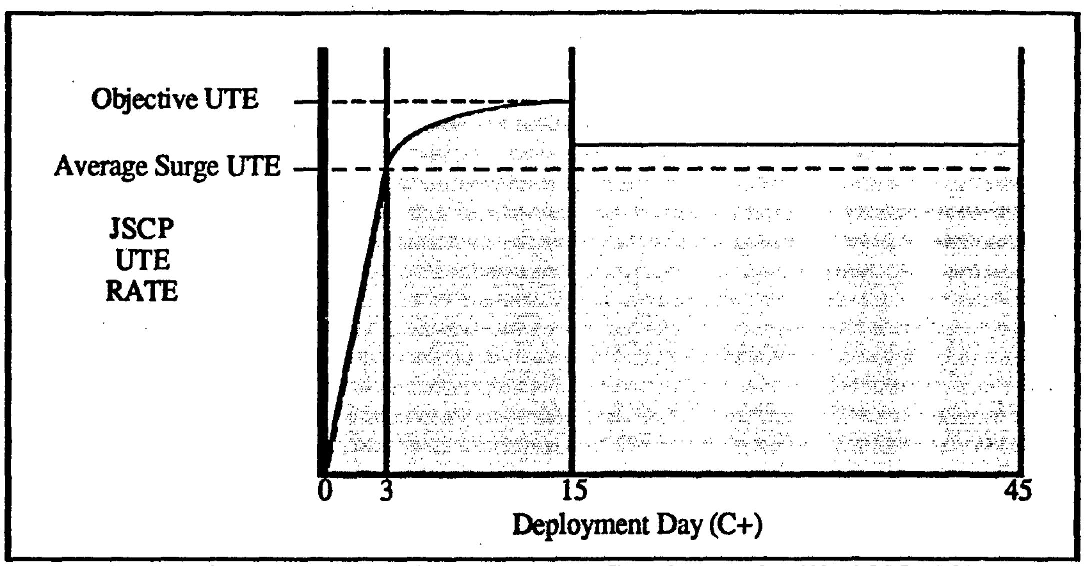


Figure 29. Surge Period UTE Rates (Gearing and Hill, 1988: 20)

(4) An overall UTE rate for the surge period is also reported (the time average of the UTE rate for each day).

UTE rates are not meant to be a constraint during an actual deployment. If higher UTE rates can be achieved, then they will be. But the published UTE rates, developed by experts using historical data and computer models, represent the highest expected utilization rates that can be achieved [Gearing and Hill, 1988:21].

6.0 MODEL INPUT

All input to the model will be included in the model formulation or will be contained in separate files built using any ASCII editor. All input parameters should be consistent in dimension (same order of magnitude) to obtain reliable results. Specific input requirements are listed next.

6.1 PLANNING HORIZON

The planning horizon of the model will be divided into equal time periods. In general, the time period used determines both the size of the model that must be solved and the accuracy of the results. A smaller unit of time (e.g. one time unit equals one hour) will result in a larger model with very detailed results. A larger unit of time (e.g. one time unit equals one day) will result in a more tractable model, but the results will be less meaningful. Thus choosing the time unit to use will require a trade-off between the level of detail desired and the response time required.

6.2 MOVEMENT REQUIREMENTS

Information on the movement requirements includes:

(1) number of military units being moved;
(2) on-load station for each unit;
(3) off-load station for each unit;
(4) amount to be moved in each cargo class;

(5) pickup window for each unit's requirements; and
(6) priority ranking of the units (optional).

6.3 ROUTE INFORMATION

Each route will be described as follows:
(1) a sequence of airfields from aircraft home field to on-load, through enroute airfields to off-load, and back to the aircraft home field;
(2) the flight time of each route leg (hours required to fly from one base in the sequence to the next);
(3) the expected ground time at each of the on-load, enroute, and off-load airfields;
(4) the cycle time for each type of aircraft on the route. Cycle time is defined to be the time required for a particular aircraft type to fly the route from home field, to on-load, enroute, and off-load airfields, and back to home field. Any delay associated with post-mission maintenance or crew-rest is added to the cycle time such that, at the end of the time, the aircraft is ready to perform a new mission.

6.4 AIRFIELD PARAMETERS

For each on-load and off-load airfield a throughput limit will be input for each time period of the model. In addition, both parking and working MOG limits will be input for each on-load, off-load, and enroute airfield in the model.

6.5 AIRCRAFT CHARACTERISTICS

The following information on the aircraft used in the airlift will be input:
(1) types available;
(2) number of each type of aircraft available at each home (operating) airfield during each time period;
(3) cycle time for each aircraft type on each possible route (see section 6.3);
(4) MOG used by each type of aircraft;
(5) capacity of each type of aircraft for its preferred cargo type;
(6) capacity of each type of aircraft for its non-preferred cargo type (if applicable); and
(7) the appropriate UTE rate for each aircraft type and surge period. UTE rates may also be enforced for a particular group of aircraft within each type. For example, the utilization of aircraft from a particular unit or base may be constrained to prevent over-utilization.

7.0 MODEL OUTPUT

The output from the model will consist of a printed solution identifying the optimal mission schedule, number and mix of aircraft used, and any shortfall.

7.1 AIRCRAFT AND MISSION INFORMATION

The model will report optimal values of two primary decision variable types. First, the shortfall for each unit and cargo class will be reported. Second, a flow variable for each aircraft type, cargo type, and time period will be reported. This flow variable represents the number of missions scheduled by the model to start in each time period.

7.2 BOTTLENECK AND SENSITIVITY INFORMATION

Information on the sensitivity of the optimal solution to changes in the input parameters will be available including:

(1) the sensitivity of the solution to changes in the "cost" of using a particular aircraft or of shortfalling a particular unit's movement requirements; and
(2) information on which of the constrained resources (aircraft and airfields) is most critical and should be the focus of any effort to add resources to improve the airlift flow.

7.3 POST-PROCESSING

A wide range of post-processing will be available after the optimal solution is found including:

(1) collecting mission information on like aircraft to report a schedule of missions to be flown by aircraft type;
(2) calculating the total number of missions scheduled;
(3) reporting the composition of the aircraft fleet used; and
(4) calculating other comparative statistics such as ton-miles or utilization.

8.0 MODEL VALIDATION

The model will not be useful unless it can routinely provide reasonably accurate solutions in a timely manner. The following criteria and tests will be used to judge the validity of the model developed.

8.1 VALIDATION CRITERIA.

As mentioned in the overview, the model may be used in two different situations. First, the model may be used in the early stages of deliberate planning. In this situation, the information available as input to the model is approximate, and the response time of the model is not critical. The second situation is the more demanding on the model. In this situation, the model is used for execution planning, when the response time is critical. Consequently, validation criteria will be used to satisfy the more demanding situation where the model is used to help in the execution planning process as follows:

(1) Ease of use. Information that is required for input to the model must be easily obtainable from sources already available in the planning process and not require extensive transformations prior to input.
(2) Response time. The model must be able to compute a solution to a problem of realistic size within a reasonable amount of time. While any quantification of this criteria is purely arbitrary, a reasonable goal may be to provide a solution to a 30 day planning problem within 30 minutes of clock time on a VAX minicomputer.
(3) Accuracy. The model must be able to determine the resources required to perform an airlift flow within plus or minus 10 percent of the actual requirements.
(4) Output. The output from the model must clearly state the optimal resource requirements, expected shortfalls, and sensitivity information.

8.2 VALIDATION STEPS.

Validation of the model will be performed in two primary ways.

(1) Face Validation. First, this specification will be reviewed by experts (airlift flow planners) to determine if it adequately represents the problem. Also, the model will be solved using a typical scenario and the results will again be scrutinized by experts. These steps are performed to compare the model against validation criteria items (1), Ease of use, and (4), Output.

(2) Second, historical data from a significant airlift operation already performed will be processed through the model. This step is performed to compare the model against validation criteria (2), Response time, and (3), Accuracy.

Appendix B: Design Document for the Airlift Capabilities Estimation Prototype (ACEP)

1.0 OVERVIEW

The Airlift Capabilities Estimation Prototype (ACEP) model is a deterministic model that provides rough estimates of the resources needed to perform an airlift operation. The original formulation was developed by Ingrid K. Busch and Michael R. Hilliard of the Operations Research Group, Center for Transportation Analysis, Oak Ridge National Laboratory. This design document is based upon updated model requirements.

2.0 APPLICABLE DOCUMENTS

2.1 Busch, Ingrid K. and Michael R. Hilliard. *An Airlift Capabilities Estimation Model for the Military Airlift Problem.* Operations Research Group, Center for Transportation Analysis, Oak Ridge National Laboratory, 1992.

2.2 Model Requirements Specification for the Airlift Capabilities Estimation Prototype, 1992.

3.0 PROBLEM SUMMARY

The airlift planning problem can be described as follows: it is desired to transport a known amount of cargo and passengers from a set of on-load airfields, through a sequence of enroute airfields, to one or more off-load airfields within a specified amount of time. A fleet of heterogeneous aircraft is assigned to perform the airlift, and a set of possible routes to move the cargo from origins to destinations is available. There are many conflicting objectives and constraints in the problem and many formulations are possible. In addition, problems of practical size and importance involve dozens of airfields, hundreds of movement requirements and aircraft, and span over a planning horizon of 90 days or more, making the resulting model computationally difficult to solve.

4.0 OVERVIEW OF MODELING APPROACH

A linear programming approach is taken so that the model may provide an optimal solution while remaining flexible and fast. Non-integer solutions are expected, but because the model is used only to obtain very rough estimates of resource requirements, this is acceptable. Depending upon how the model is implemented and the speed of the linear program (LP) solver used, the model has the potential to be very flexible. Once a solution is found for a particular set of inputs, changes can be made to the data and the model can be restarted from a previous solution. An additional advantage of the linear programming approach is that required sensitivity analysis information is automatically calculated.

5.0 MODEL ASSUMPTIONS

Some assumptions are required in order to use a linear programming approach. Most notably, it is assumed that the non-integer (i.e. infeasible) solution can be used to provide the decision information required. Some other assumptions that are made are:

(1) The objective function and constraints in the problem can be modeled with linear equations. The classic assumptions of linear programming are:

- proportionality – the contribution of each decision variable is proportional to the value of the variable.
- additivity – the contribution of each variable is independent of the values of the other variables.
- divisibility – the decision variables are allowed to take on non-integer values.
- certainty – each parameter in the problem is known with certainty.
- nonnegativity – all decision variables are restricted to values greater than or equal to zero.

(2) Non-preferred cargo carried by aircraft will be of type "bulk" or "passengers" only. It is unlikely that any aircraft type will be able to carry "outsize" cargo in a non-preferred capacity.

(3) Aircraft are assumed to "consume" both parking and working MOG at each airfield in proportion to the size of the aircraft and the number being serviced. For example, if a single C-141 is considered to take 1.0 unit of MOG, then two C-141s will take 2.0 units.

6.0 MATHEMATICAL FORMULATION

The mathematical formulation presented here is based upon the formulation of Busch and Hilliard with additions and modifications where necessary to meet the model specifications.

6.1 PARAMETERS

The following symbols are used to represent model parameters or input data:

b is the set of airfields (on-load, enroute, and off-load) considered in the model.

r is the set of routes available from on-load airfields to off-load airfields.

t is the planning horizon of the deployment in time units.

u is the customers (military units) that require airlift support from a particular on-load airfield to a particular off-load airfield (also called a movement requirement).

p is the aircraft types available (e.g. C-5, C-141, P-747, etc.).

c is the classes of cargo where $c \in \{0 = \text{outsize}, 1 = \text{bulk}, 2 = \text{passengers}\}$. The class "oversize" can also be included if desired.

Other model parameters are explained following the objective function or constraint equation where they are first used.

6.2 MODELING OBJECTIVE

The objective of the model is to minimize the total "cost" of the operation where the two primary components of cost are associated with shortfall (undelivered cargo) and use of aircraft resources. This objective provides a flexible way of controlling the model to minimize shortfall, aircraft usage, or any suitable combination.

6.2.1 DECISION VARIABLES

Values for the following variables are determined by the ACEP model:

X^t_{upcr} is the number of aircraft of type p scheduled to begin an airlift mission on day t carrying cargo of class c from unit u on route r. This variable can also be interpreted as the flow (movement) of preferred cargo of class c from unit u along route r where amount of cargo moved is X payloads of aircraft type p.

Y^t_{upcr} is the opportune movement of non-preferred cargo of class c from unit u carried on aircraft type p along route r beginning on day t.

Z_{uc} is shortfall – the amount of cargo of class c from unit u left undelivered at the end of the planning horizon.

6.2.2 OBJECTIVE FUNCTION

The objective function used in the basic problem is to minimize the "cost" of the airlift. This cost is defined to be a combination of the penalty cost of not delivering requirements (shortfalls) and the cost of operating the aircraft. This objective function is obviously not meant to be an accurate indicator of the true cost of the operation, which would be almost impossible to determine in the early stages, but is meant to combine the conflicting objectives of minimizing shortfall and the number of aircraft employed in a way that will result in a balanced airlift flow. Mathematically, the objective function is expressed as:

$$\text{Minimize} \quad \Sigma_{uc}\rho_{uc}Z_{uc} + \Sigma_{ucprt}v^t_{pu}X^t_{upcr} \tag{1}$$

where: ρ_{uc} is the penalty for shortfalling one unit of cargo class c of requirement u.

v^t_{pu} is the cost of using aircraft type p to move a requirement from unit u during time period t.

By controlling the values of ρ and ν, the objective function can be formulated to minimize either shortfall (by setting all $\nu = 1$ and adjusting ρ), aircraft usage (by setting all $\rho = 1$ and adjusting ν), or both using a suitable combination of weights. The model can be made to deliver each movement requirement as early as possible within the pickup window by increasing ν as t increases.

6.3 CONSTRAINT EQUATIONS

The constraints used in ACEP can be grouped into four categories: movement constraints, aircraft constraints, throughput constraints, and enroute constraints.

6.3.1 PRIMARY MOVEMENT CONSTRAINTS

The following constraints ensure that all cargo and passengers are either delivered or reported as shortfall. Equation (2) represents outsize cargo, equation (3) represents bulk cargo, and equation (4) is for passengers. One set of equations is necessary for each requirement. It is assumed that no aircraft types have a non-preferred cargo type of "outsize".

$$\sum_{p\in\Omega_0}\sum_{r\in C(p,b)}\sum_{t\in\Delta_{urp}}\gamma_{rp}X^{t}_{up0r} + Z_{u0} = q_{u0} \quad (2)$$

$$\sum_{p\in\Omega_1}\sum_{r\in C(p,b)}\sum_{t\in\Delta_{urp}}\gamma_{rp}X^{t}_{up1r} + \sum_{p\in\Omega_1}\sum_{r\in C(p,b)}\sum_{t\in\Delta_{urp}}\phi_{rp}Y^{t}_{up1r} + Z_{u1} = q_{u1} \quad (3)$$

$$\sum_{p\in\Omega_2}\sum_{r\in C(p,b)}\sum_{t\in\Delta_{urp}}\gamma_{rp}X^{t}_{up2r} + \sum_{p\in\Omega_2}\sum_{r\in C(p,b)}\sum_{t\in\Delta_{urp}}\phi_{rp}Y^{t}_{up2r} + Z_{u2} = q_{u2} \quad (4)$$

where: q_{uc} is the amount (in tons or number of passengers) of cargo in class c of requirement u.

Ω_c is the set of aircraft types with capability to carry cargo type c.

C(p,b) is the set of routes that can be flown by aircraft type p based at airfield b.

γ_{rp} is the capacity of aircraft type p on route r for its preferred cargo type.

ϕ_{rp} is the capacity of aircraft type p on route r for its non-preferred cargo.

Δ_{urp} is the pickup window (set of feasible time units for loading aircraft) for requirement u using aircraft type p on route r.

6.3.2 SECONDARY MOVEMENT CONSTRAINTS

These constraints ensure that the opportune movement of cargo and passengers (non-preferred types) is proportional to the movement of preferred cargo (one equation for each route, aircraft type, and time unit). Equation (5) represents bulk cargo carried on passenger aircraft and equation (6) represents passengers carried on cargo aircraft. Again, it is assumed that any non-preferred capability will be in cargo classes "bulk" or "passengers" only.

$$\sum_{u}Y^{t}_{up1r} \le \sum_{u}X^{t}_{up2r} \quad (5)$$

$$\sum_{u}Y^{t}_{up2r} \le \sum_{u}[X^{t}_{up1r} + X^{t}_{up0r}] \quad (6)$$

6.3.3 AIRCRAFT CONSTRAINTS

The constraints represented in equation (7) ensure that the number of aircraft used is less than or equal to the number available (one equation for each aircraft type p based out of airfield b during time t).

$$\sum_{uc}\sum_{r\in C(p,b)}\sum^{t}_{k=(t-\alpha_{pr})}X^{k}_{upcr} \le a^{t}_{pb} \quad (7)$$

where: α_{pr} is the length of the round trip (cycle time) for aircraft type p on route r. Note: must be an integer value.

a^{t}_{pb} is the number of aircraft of type p based at airfield b on day t.

Equation (8) ensures that the aircraft UTE rates are not exceeded. One equation is required for each UTE period, aircraft type, and operating base.

$$\Sigma_{t\in\mu}\Sigma_{ucr}\Sigma_{b\in R(b,t-\sigma_{rb}-1)}F_{rbp}X^{t-\sigma_{rb}-1}_{upcr} \quad \leq \quad \Sigma_{t\in\mu}UTE^{\mu}_{p}a^{t}_{pb} \qquad (8)$$

where: μ is the set of time units (surge period) over which the UTE is enforced. These will normally be (1,2,3), (1, ..., 15), and (1, ..., 45).

$R(b,t)$ is the set of airfields visited t time units after the start of route r.

σ_{rb} is the number of time units required to reach airfield b on route r.

F_{rbp} is the time required for an aircraft of type p to fly to the next airfield on route r after airfield b.

UTE^{μ}_{p} is the objective UTE rate for aircraft type p over surge period μ (daily UTE rate adjusted for the time units used).

6.3.4 THROUGHPUT CONSTRAINTS

The throughput constraints ensure that the airlift operation does not exceed the capabilities of the aerial port units at the on-load and off-load airfields (one equation for each on-load or off-load airfield and time unit). Equation (9) is for cargo (bulk and outsize) throughput and (10) is for passenger throughput.

$$\begin{aligned}&\Sigma_{u\ni o(u)=b}[\Sigma_{p\in\Omega_1}\Sigma_{r\in C(p,b)}\gamma_{rp}(X^{t}_{up0r}+X^{t}_{up1r}) + \Sigma_{p\in\Omega_2}\Sigma_{r\in C(p,b)}\phi_{rp}Y^{t}_{up1r}] + \\ &\Sigma_{u\ni d(u)=b}[\Sigma_{p\in\Omega_1}\Sigma_{r\in C(p,b)}\gamma_{rp}(X^{t-\sigma_{rb}-1}_{up0r}+X^{t-\sigma_{rb}-1}_{up1r}) + \\ &\Sigma_{p\in\Omega_2}\Sigma_{r\in C(p,b)}\phi_{rp}Y^{t-\sigma_{rb}-1}_{up1r}] \qquad \leq \theta^{t}_{b1} \qquad (9)\end{aligned}$$

$$\begin{aligned}&\Sigma_{u\ni o(u)=b}[\Sigma_{p\in\Omega_2}\Sigma_{r\in C(p,b)}\gamma_{rp}X^{t}_{up2r} + \Sigma_{p\in\Omega_1}\Sigma_{r\in C(p,b)}\phi_{rp}Y^{t}_{up2r}] + \\ &\Sigma_{u\ni d(u)=b}[\Sigma_{p\in\Omega_2}\Sigma_{r\in C(p,b)}\gamma_{rp}X^{t-\sigma_{rb}-1}_{up2r} + \Sigma_{p\in\Omega_1}\Sigma_{r\in C(p,b)}\phi_{rp}Y^{t-\sigma_{rb}-1}_{up2r}] \leq \theta^{t}_{b2} \qquad (10)\end{aligned}$$

where: $\theta^{t}_{bc'}$ is the throughput capacity of airfield b at time t for cargo class c' where $c' \in \{1 = \text{cargo}, 2 = \text{passengers}\}$.

$o(u)$ is the on-load airfield for requirement u.

$d(u)$ is the off-load airfield for requirement u.

6.3.5 ENROUTE CONSTRAINTS

The enroute constraints ensure that the capacity of the enroute airfields to support the airlift (MOG) is not exceeded (one equation for each enroute airfield and time unit).

$$\Sigma_{ucp}\Sigma_{r\in C(p,b)\cap R(b,t-\sigma_{rb}-1)} m_{bp} X^{t-\sigma_{rb}-1}{}_{upcr} \quad \leq \quad \min(e^{t}{}_{b}, w^{t}{}_{b}) \qquad (11)$$

where: m_{bp} is the amount of MOG (parking or working – depending on which is most limiting at the airfield) used by aircraft type p at airfield b.

$e^{t}{}_{b}$ is the ramp capacity (parking MOG) of airfield b at time t.

$w^{t}{}_{b}$ is the working MOG of airfield b at time t.

6.4 DELIVERED CARGO-TO-PASSENGER RATIO

Historically, one ton of cargo must be delivered for each passenger. This constraint ensures that the correct proportion of cargo to passengers is delivered over the planning period. Since the 1 to 1 ratio is not always appropriate, the ratio can be adjusted to meet planning needs.

$$\Sigma_{tupr}\gamma_{rp}X^{t}{}_{up0r} + \Sigma_{tupr}\left[\gamma_{rp}X^{t}{}_{up1r} + \phi_{rp}Y^{t}{}_{up1r}\right] \geq \beta * \Sigma_{tupr}\left[\gamma_{rp}X^{t}{}_{up2r} + \phi_{rp}Y^{t}{}_{up2r}\right] \qquad (12)$$

where: β is the desired ratio of cargo to passengers delivered.

6.5 MINIMUM LOAD

All the decision variables are defined to be non-negative. A minimum load to justify scheduling a mission by a particular aircraft type may be enforced as follows:

$$X^{t}{}_{upcr} \geq \pi^{t}{}_{upcr} * \text{minload}_{pc} \qquad (13)$$

where: $\pi^{t}{}_{upcr}$ is $\begin{cases} 1 \text{ if } X^{t}{}_{upcr} \text{ is greater than zero} \\ 0 \text{ otherwise} \end{cases}$

minload_{pc} is the minimum load desired as a percentage of capacity.

Note, however, that this constraint requires a reformulation of the objective function to include the 0 – 1 variables and results in a mixed-integer programming problem of very large size.

7.0 MODEL OUTPUT

The output from the model will consist of a printed solution identifying the optimal value found for the objective function and decision variables as well as sensitivity analysis information associated with each variable and constraint.

7.1 DECISION VARIABLES

The model will report optimal values of two primary decision variable types. First, the shortfall for each unit and cargo class will be reported. Second, a flow variable for each aircraft type, cargo type, and time period will be reported. This flow variable may be interpreted as the number of sorties scheduled by the model to start in each time period. This variable will NOT be integer in most cases.

7.2 SENSITIVITY INFORMATION

The standard solution sensitivity information provided by linear program solvers will be available including:

(1) the reduced cost for each non-basic decision variable (the expected improvement in the objective function coefficient associated with the variable that must be made in order for the solution to include the variable at a non-zero level). This information can be used to determine the amount by which the cost of using a particular aircraft type must improve in order to use that type during a time period when it is not currently included in the optimal solution.

(2) the shadow price for each constraint (the expected improvement in the objective function given by a unit increase in availability of the resource being constrained). This information can be used to determine which of the constrained resources (aircraft and airfields) is most critical and should be the focus of any effort to add resources to improve the airlift flow.

The sensitivity information outlined above is unique for the particular optimal solution found by the solver. If the solution (optimal basis) changes, then the sensitivity information must be recomputed.

8.0 VERIFICATION OF DESIGN

Table 21 shows a trace of the model requirements to the mathematical equation in the formulation where the requirement is satisfied:

TABLE 21

Requirements Verification Matrix

	Model Equation that Satisfies Requirement												
Requirement	1	2	3	4	5	6	7	8	9	10	11	12	13
Section 4.1													
Minimize Shortfall	X												
Section 4.2													
Maximize Efficiency	X												
Section 5.1													
Planning Horizon	X	X	X	X	X	X	X	X	X	X	X	X	X
Section 5.2													
Deliver Outsize		X											
Deliver Bulk			X										
Deliver Passengers				X									
Use Delivery Window		X	X	X									
Cargo/Passenger Ratio												X	
Section 5.3													
Available Routes		X	X	X									
Section 5.4													
Cargo Throughput Limits									X				
Passenger Throughput Limits										X			
MOG Limits											X		
Section 5.5													
Aircraft Capacity		X	X	X	X	X							
Aircraft Availability							X						
Minimum Load													X
Preferred Cargo		X	X	X									
Non-preferred Cargo					X	X							
UTE Rates								X					

Appendix C: GAMS Implementation of ACEP with Contrived Scenario

```
******************************************************************
* Filename:   ACEP.gms                                           *
*                                                                *
* Purpose:    Verification/Validation of Airlift Capability      *
*             Estimation Prototype (ACEP) Model.                 *
*                                                                *
******************************************************************
$OFFSYMXREF
$OFFSYMLIST
$OFFUPPER
OPTIONS ITERLIM=5000,LIMCOL=0,LIMROW=0 ;
OPTION SOLPRINT=OFF ;
**************************************************************
* time =      set of time units in model planning period     *
*                                                            *
* NOTE:       set should cover C01 to end of desired horizon *
*             plus enough time units to cover the longest route *
*             cycle.  This prevents the model from over-     *
*             scheduling at the end.                         *
**************************************************************

SET time          Time periods in the model
    /C01 * C64 /

 alias(time,time2) ;

**************************************************************
* TUNIT =     time units per 24 hour period (e.g. TUNIT = 2  *
*             means that 24/2 = 12 hour time periods are used) *
**************************************************************

SCALAR TUNITS / 2 /;

**************************************************************
* unit =      designator for the military units (airlift     *
*             customers) that require transport.             *
**************************************************************

SET unit          Unit requiring movement of cargo or pax
    / U01 * U04 / ;

**************************************************************
* plane =     set of aircraft resources available for use at *
*             any time during the airlift operation.         *
**************************************************************
```

```
SET plane        Types of aircraft available
    / C141, C005, C017, P747 /

************************************************************
* class =     set of cargo classes (usually the airlift    *
*             requirements can be classified into AT MOST four *
*             classes: outsize, oversize, bulk and passengers) *
************************************************************

SET class        Types of cargo or pax
    / out, blk, pax / ;

************************************************************
* cargo =     set of classes that include only true cargo.  *
************************************************************

SET cargo(class)    Types of cargo only
    / out, blk / ;

************************************************************
* base =      set of airfields available for use at any time *
*             during the airlift operation. The use of the   *
*             four letter ICAO designator works well.        *
************************************************************

SET base         Airfields included in the model
    / K001 * K006 / ;

************************************************************
* route =     set of routes available for moving cargo/pax. *
*             A typical route might be:                     *
*             STOP   PURPOSE                                *
*               1    Aircraft home (operating) base         *
*               2    On-load airfield                       *
*               3    Enroute airfield(s)                    *
*               4    Off-load airfield                      *
*               5    Recovery airfield (refuel,mx)          *
*               6    Enroute airfield(s)                    *
*               7    Aircraft home base                     *
************************************************************

SET route        Routes by which cargo can move
    / R001 * R003 /;

************************************************************
* vtinc =     set of time units for referencing days within *
*             each aircraft mission. Set should be sequential *
*             from 0 to number of time units in the longest *
*             roundtrip cycle.                              *
************************************************************
```

```
SET vtinc        Time units in the longest cycle
   /0*3/;

*********************************************************
* delta =     defines the pickup window for each requirement.     *
*             Delta is indexed by route and aircraft type to      *
*             provide further flexibility in controlling the      *
*             scheduling of missions. Note that MANY of the       *
*             constraints equations are $ controlled with the     *
*             delta set -- so tighter and more specific delta     *
*             windows should speed up the solve time.             *
*********************************************************

SET delta(unit,route,plane,time)           Feasible pick-up times
   / U01.R001.(C005,C141).(C01 * C60)
     U01.R002.C017.(C01 * C60)
     U01.R002.C141.(C05 * C60)
     U01.R003.P747.(C05 * C60)
     U02.R001.(C005,C141).(C15 * C60)
     U02.R002.(C141,C017).(C15 * C60)
     U02.R003.P747.(C15 * C60)
     U03.R001.(C005,C141).(C29 * C60)
     U03.R002.(C141,C017).(C29 * C60)
     U03.R003.P747.(C29 * C60)
     U04.R001.C141.(C01 * C60)
     U04.R002.(C141,C017).(C01 * C60) / ;

*********************************************************
* home =      defines the home (operating) bases for the          *
*             aircraft for enforcing UTE rates by base.           *
*********************************************************

SET home(base,plane)                       Home base of aircraft
   / K001.(C005,C141)
     K002.(C017,C141)
     K003.P747 / ;

*********************************************************
* head =      the starting airfield on each route.                *
*********************************************************

SET head(base,route)                       Starting point of routes
   / K001.R001
     K002.R002
     K003.R003 / ;

*********************************************************
* AMOUNT = the amount of cargo (tons) or passengers from          *
*             unit that must be moved.                            *
*********************************************************
```

```
TABLE AMOUNT(unit,class)                Amounts to be moved
          out     blk     pax
    U01   740     980    6000
    U02   440     680    7000
    U03   190    1040    1800
    U04           880    7700  ;

**********************************************************
* TCLASS = total amount to be moved in each class.       *
**********************************************************

PARAMETER TCLASS(class);
    TCLASS(class) = SUM(unit, AMOUNT(unit,class));
DISPLAY TCLASS;

**********************************************************
* RATIO =    the desired ratio of cargo to passengers delivered.   *
*            Historically, the ratio is often near 1.0.            *
**********************************************************

SCALAR RATIO / 0.90 /;

**********************************************************
* flytime =    flying time (in hours) accumulated in each          *
*              time period (vtinc) on each route.  Note that       *
*              aircraft are assumed to start their missions        *
*              at the beginning of vtinc 0.                        *
**********************************************************

PARAMETER flytime(route,vtinc)          Flight time between stops
    / R001.0  6, R001.1  8, R001.2  9, R001.3  6
      R002.0  8, K002.1  7, R002.2  8, R002.3  9
      R003.0 10, R003.1  5, R003.2 10, R003.3  3  /;

**********************************************************
* legtime =    time units on each route (vtinc) when the           *
*              aircraft will be located at each airfield on        *
*              the route.                                          *
**********************************************************

SET legtime(route,base,vtinc)           Time periods at intermediate stops
    / R001.(K001,K003,K004).0
      R001.(K004,K005).1
      R001.(K006).2
      R001.(K004,K001).3
      R002.(K002,K003).0
      R002.(K004,K005).1
      R002.(K006).2
      R002.(K004,K002).3
      R003.(K003,K004).0
      R003.(K005,K006).1
    R003.(K004).2
    R003.(K003).3    / ;

**********************************************************
* onload =    the onload airfield for each requirement unit.         *
**********************************************************

SET onload(unit,base)                     Onload point for each unit
    / (U01,U02,U03,U04).K003 / ;

**********************************************************
* offload =    the offload airfield for each requirement unit.        *
**********************************************************

SET offload(unit,base)                    Offload point for each unit
    / (U01,U02,U03,U04).K005 / ;

**********************************************************
* numplane = the number of each type of aircraft at each            *
*                operating base available for use during each        *
*                time period.                                         *
**********************************************************

PARAMETER numplane(plane,base,time) Number of aircraft available
    / C141.K001.(C01*C64)          10
      C141.K002.(C01*C64)           5
      C005.K001.(C01*C14)          10
      C005.K001.(C15*C64)           8
      C017.K002.(C01*C64)           7
      P747.K003.(C05*C64)           3  / ;

**********************************************************
* period =    the UTE periods to be tracked in the model.            *
**********************************************************

SET period        UTE Periods tracked by model
    / FIRST, SECOND, THIRD / ;

**********************************************************
* UTEPRD = definition of the UTE period (time units                  *
*               covered).                                            *
**********************************************************

SET UTEPRD(period,time)                   Time periods covered by UTE periods
    / FIRST.(C01 * C06)
      SECOND.(C01 * C30)
      THIRD.(C01 * C64) / ;

**********************************************************
* UTENUM = number of time units in each UTE period.                  *
**********************************************************
```

```
PARAMETER UTENUM(period)          Number of Time Units in each UTE Period
   / FIRST       6
     SECOND    30
     THIRD      64  /;

************************************************************
* UTERATE = the DAILY UTE rate for each aircraft type (in          *
*                  hours of flight time per day).                                *
************************************************************

TABLE UTERATE(period,plane) Daily UTE rates for aircraft
                C141      C005       C017      P747
     FIRST     4            4            4
     SECOND 13          12          14           10
     THIRD    12          10          12           10  ;

************************************************************
* UTEHOURS = the total pool of flying hours available for          *
*                   each UTE period, aircraft type and operating        *
*                   base (computed automatically).                            *
************************************************************

PARAMETER UTEHOURS(period,base,plane);
     UTEHOURS(period,base,plane)$HOME(base,plane) =
     SUM(time$(UTEPRD(period,time)$NUMPLANE(plane,base,time)),
     NUMPLANE(plane,base,time) * (UTERATE(period,plane)/TUNITS) );

************************************************************
* CYCLE =   the number of time units required to complete             *
*                 each route by each type of aircraft.                          *
************************************************************

TABLE CYCLE(route,plane) Length of roundtrip cycle
                C141      C005       C017      P747
     R001     4            4            4            4
     R002     4            4            4            4
     R003     4            4            4            4  ;

************************************************************
* PARK =     the number of aircraft parking spots available              *
*                 at each airfield and unit of time.  NOTE:  the               *
*                 time index is included to account for other                  *
*                 airlift traffic that may effect ramp space                   *
*                 available at various times.                                     *
************************************************************

PARAMETER PARK(base,time)          Parking MOG limits at each base
   / K001.(C01*C64)    25
     K002.(C01*C64)    75
     K003.(C01*C64)    15
     K004.(C01*C10)    30
     K004.(C11*C64)   40
     K005.(C01*C64)   25
     K006.(C01*C64)   20  /;

**************************************************************
* WORK =   the working MOG limits at each airfield and time.        *
*               Working MOG includes such factors as refueling              *
*               capability and operating hours.                                   *
**************************************************************

PARAMETER WORK(base,time)          Working MOG limits at each base
   / K001.(C01*C64)   50
     K002.(C01*C64)   25
     K003.(C01*C64)   20
     K004.(C01*C64)   25
     K005.(C01*C64)   50
     K006.(C01*C04)   20
     K006.(C05*C20)   10
     K006.(C21*C64)   20  /;

**************************************************************
* AMOG =   the most constraining (i.e. minimum) MOG at each          *
*               airfield and time.                                                  *
**************************************************************

PARAMETER AMOG(base;time);
     AMOG(base,time) = MIN(PARK(base,time),WORK(base,time));

**************************************************************
* MOG =      the amount of parking and working MOG that each          *
*               type of aircraft uses at each airfield.                        *
*               Note:  larger aircraft will use a proportionally            *
*               higher amount of MOG.  The airfield index is               *
*               included to provide further flexibility.                      *
**************************************************************

TABLE MOG(base,plane)  MOG used by each aircraft type at each base
              C141      C005      C017       P747
     K001       1          2        1.2         2.1
     K002       1          2        1.2         2.1
     K003       1          2        1.2         2.1
     K004       1          2        1.2         2.1
     K005       1          2        1.2         2.1
     K006       1          2        1.2         2.1   ;

**************************************************************
* THRU =    the cargo and passenger throughput limitations at        *
*               the onload and offload airfields (expressed in              *
*               tons of cargo or number of passengers per time            *
*               unit).                                                                  *
**************************************************************
```

```
PARAMETER THRU(base,time,class)   Throughput limits at onload & offload
   / K003.(C01*C64).pax   2500
     K003.(C01*C05).blk    120
     K003.(C06*C64).blk    180
     K005.(C01*C08).pax   1000
     K005.(C09*C64).pax   3000
     K005.(C01*C64).blk    300  /;

************************************************************
* RHO =      the penalty associated with failing to deliver one          *
*            ton of cargo or one passenger from each unit.               *
*            Note that unit priorities can be handled by                 *
*            adjusting this parameter.  Also, RHO should be set          *
*            greater than or equal to the highest penalty for            *
*            each sortie (see NU).                                       *
************************************************************

TABLE RHO(unit,class)  Penalty for shortfall
           out      blk      pax
     U01   100      100      100
     U02   100      100      100
     U03   100      100      100
     U04   100      100      100   ;

************************************************************
* NU =       the penalty associated with using each aircraft             *
*            type to deliver cargo/pax from each unit.  Note:            *
*            the use of a particular type of aircraft can be             *
*            controlled (minimized or maximized) by setting the          *
*            penalty appropriately hign or low in relation to            *
*            the other aircraft penalties.                               *
************************************************************

TABLE NU(unit,plane)  Penalty for each sortie flown
              C141      C005      C017      P747
     U01        1         1         1         2
     U02        1         1         1         2
     U03        1         1         1         2
     U04        1         1         1         2   ;

************************************************************
* NUNU =     the aircraft use penalty adjusted to increase               *
*            with time.  This forces the model to move the               *
*            cargo/pax as soon as possible (maximizes flow               *
*            until all requirements are delivered).                      *
************************************************************

PARAMETER NUNU(unit,plane,time);
     NUNU(unit,plane,time) = NU(unit,plane) + ORD(time) ;
```

```
***********************************************************
* PREFER = the cargo or passenger capacity for each       *
*          aircraft type for it's preferred cargo (in tons *
*          or number of passengers).                       *
***********************************************************

PARAMETER PREFER(plane,class,route) Capacities for preferred cargo
   / C141.blk.(R001,R002)   19
     C141.pax.(R001,R002)  130
     C005.(out,blk).R001    60
     C017.out.R002          38
     C017.blk.R002          40
     P747.pax.R003         365  /;

***********************************************************
* NONPREF = the cargo or passenger capacity for each      *
*          aircraft type for it's non-preferred cargo (in  *
*          tons or number of passengers).                  *
***********************************************************

PARAMETER NONPREF(plane,class,route) Capacities for nonpreferred cargo
    / C005.pax.R001  68 / ;

FREE VARIABLES
COBJ                                  Cost of operation
DOBJ                                  Amount of tons & pax delivered
;
POSITIVE VARIABLES
X(time,unit,plane,class,route)        Number of sorties
Y(time,unit,plane,class,route)        Opportune movement of nonpreferred cargo
Z(unit,class)                         Amount left undelivered (shortfall)
;

EQUATIONS
COST                                  Objective "cost" function
DELIVER                               Objective flow function
DELIVPAX(unit)                        Account for passenger requirements
DELIVBLK(unit)                        Account for bulk cargo requirements
DELIVOUT(unit)                        Account for outsize cargo requirements
NONPREF1(route,plane,time)            Opportune cargo on pax planes
NONPREF2(route,plane,time)            Opportune pax on cargo planes
AVAIL(plane,base,time)                Number of aircraft available
THRUPUT1(base,time)                   Cargo throughput at onload & offload airfields
THRUPUT2(base,time)                   Pax throughput at onload & offload airfields
MOGLIM(base,time)                     MOG limits at enroute airfields
UTELIM(period,base,plane)             Aircraft utilization constraints
DELRATO                               Ratio of delivered cargo to pax
;
```

```
COST.. COBJ =E= SUM((unit,class)$AMOUNT(unit,class), RHO(unit,class) *
      Z(unit,class)) + SUM((unit,class,route,plane,time)
      $(delta(unit,route,plane,time)$PREFER(plane,class,route)),
      NUNU(unit,plane,time) * X(time,unit,plane,class,route) ) ;

DELIVER.. DOBJ =E= SUM((time,unit,plane,class,route)
      $delta(unit,route,plane,time),
      PREFER(plane,class,route) * X(time,unit,plane,class,route)
      + NONPREF(plane,class,route) * Y(time,unit,plane,class,route) ) ;

DELIVPAX(unit)$AMOUNT(unit,"pax").. AMOUNT(unit,"pax") =E= Z(unit,"pax")
      + SUM((route,plane,time)$delta(unit,route,plane,time),
      PREFER(plane,"pax",route) * X(time,unit,plane,"pax",route)
      + NONPREF(plane,"pax",route) * Y(time,unit,plane,"pax",route) ) ;

DELIVBLK(unit)$AMOUNT(unit,"blk").. AMOUNT(unit,"blk") =E= Z(unit,"blk")
      + SUM((route,plane,time)$delta(unit,route,plane,time),
      PREFER(plane,"blk",route) * X(time,unit,plane,"blk",route)
      + NONPREF(plane,"blk",route) * Y(time,unit,plane,"blk",route) ) ;

DELIVOUT(unit)$AMOUNT(unit,"out").. AMOUNT(unit,"out") =E= Z(unit,"out")
      + SUM((route,plane,time)$delta(unit,route,plane,time),
      PREFER(plane,"out",route) * X(time,unit,plane,"out",route) ) ;

NONPREF1(route,plane,time)$NONPREF(plane,"blk",route)..
      SUM(unit$delta(unit,route,plane,time),
      X(time,unit,plane,"pax",route)$PREFER(plane,"pax",route)
      - Y(time,unit,plane,"blk",route)$NONPREF(plane,"blk",route) ) =G= 0.0;

NONPREF2(route,plane,time)$NONPREF(plane,"pax",route)..
      SUM(unit$delta(unit,route,plane,time).
      X(time,unit,plane,"out",route)$PREFER(plane,"out",route)
      + X(time,unit,plane,"blk",route)$PREFER(plane,"blk",route )
      - Y(time,unit,plane,"pax",route)$NONPREF(plane,"pax",route) ) =G= 0.0;

AVAIL(plane,base,time)$NUMPLANE(plane,base,time)..
      NUMPLANE(plane,base,time) =G= SUM((route,time2)
      $((ORD(time2) le ORD(time)) and (ORD(time2) gt (ORD(time)
      - CYCLE(route,plane)))$head(base,route)),
      SUM((unit,class)$(delta(unit,route,plane,time2)
      $PREFER(plane,class,route)), X(time2,unit,plane,class,route) ) ) ;

THRUPUT1(base,time)$THRU(base,time,"blk").. THRU(base,time,"blk") =G=
      SUM(unit$onload(unit,base),
      SUM((route,plane)$delta(unit,route,plane,time),
      PREFER(plane,"blk",route) * X(time,unit,plane,"blk",route) +
      PREFER(plane,"out",route) * X(time,unit,plane,"out",route) +
      NONPREF(plane,"blk",route) * Y(time,unit,plane,"blk",route) ) ) +

      SUM(unit$offload(unit,base),
      SUM((route,plane,vtinc) $(legtime(route,base,vtinc)
     $delta(unit,route,plane,time-(ORD(vtinc)-1))),
     PREFER(plane,"blk",route) *
     X(time-(ORD(vtinc)-1),unit,plane,"blk",route) +
     PREFER(plane,"out",route) *
     X(time-(ORD(vtinc)-1),unit,plane,"out",route) +
     NONPREF(plane,"blk",route) *
     Y(time-(ORD(vtinc)-1),unit,plane,"blk",route) ) ) ;

THRUPUT2(base,time)$THRU(base,time,"pax").. THRU(base,time,"pax") =G=
     SUM(unit$onload(unit,base),
     SUM((route,plane)$delta(unit,route,plane,time),
     PREFER(plane,"pax",route) * X(time,unit,plane,"pax",route) +
     NONPREF(plane,"pax",route) * Y(time,unit,plane,"pax",route) ) ) +

     SUM(unit$offload(unit,base),
     SUM((route,plane,vtinc) $(legtime(route,base,vtinc)
     $delta(unit,route,plane,time-(ORD(vtinc)-1))),
     PREFER(plane,"pax",route) *
     X(time-(ORD(vtinc)-1),unit,plane,"pax",route) +
     NONPREF(plane,"pax",route) *
     Y(time-(ORD(vtinc)-1),unit,plane,"pax",route) ) ) ;

MOGLIM(base,time)$AMOG(base,time).. AMOG(base,time) =G=
     SUM((unit,class,route,plane,vtinc)$(legtime(route,base,vtinc)
     $(delta(unit,route,plane,time-(ORD(vtinc)-1))
     $PREFER(plane,class,route))),
     MOG(base,plane) * X(time-(ORD(vtinc)-1),unit,plane,class,route) ) ;

UTELIM(period,base,plane)$HOME(base,plane)..
     UTEHOURS(period,base,plane) =G= SUM(time$UTEPRD(period,time),
     SUM((unit,class,route,vtinc)$(FLYTIME(route,vtinc)
     $(DELTA(unit,route,plane,time-(ORD(vtinc)-1))
     $(PREFER(plane,class,route)$HEAD(base,route)))),
     FLYTIME(route,vtinc) * X(time-(ORD(vtinc)-1),unit,plane,class,route)));

DELRATO.. SUM((time,unit,plane,class,route)
     $(DELTA(unit,route,plane,time)$(AMOUNT(unit,class)$cargo(class))),
     PREFER(plane,class,route) * X(time,unit,plane,class,route) +
     NONPREF(plane,class,route) * Y(time,unit,plane,class,route) )
     - RATIO * SUM((time,unit,plane,route)$DELTA(unit,route,plane,time),
     PREFER(plane,"pax",route) * X(time,unit,plane,"pax",route) +
     NONPREF(plane,"pax",route) * Y(time,unit,plane,"pax",route) ) =G= 0.0 ;

MODEL ACEPC  /  COST, DELIVPAX, DELIVBLK, DELIVOUT,
                NONPREF1, NONPREF2, MOGLIM, THRUPUT1,
                THRUPUT2, AVAIL, UTELIM, DELRATO /;

MODEL ACEPD  /  DELIVER, NONPREF1, NONPREF2, AVAIL,
                MOGLIM, UTELIM, THRUPUT1, THRUPUT2,
                DELRATO /;
```

```
************************************************************
************************************************************
* Solve model ACEPC (minimize "cost")...                   *
************************************************************
************************************************************

SOLVE ACEPC USING LP MINIMIZING COBJ ;

************************************************************
* Compute max percent of available aircraft used and the   *
* marginal value for maxed aircraft...                     *
************************************************************

PARAMETER PAVAIL(plane,base) Max Percent of Available ACFT Used;
    PAVAIL(plane,base) = 0;
PARAMETER AAVAIL(period,base, plane) Average Number ACFT Used;
    AAVAIL(period,base, plane) = SUM(day$UTEDAYS(period,day),
    -AVAIL.L(plane,base,day) / UTENUM(period) ;
PARAMETER MAVAIL(plane,base) Marginal Value of Aircraft;
    MAVAIL(plane,base) = 0;
LOOP(time,
    PAVAIL(plane,base)$NUMPLANE(plane,base,time) =
        MAX( PAVAIL(plane,base), (-AVAIL.L(plane,base,time) /
            NUMPLANE(plane,base,time)) ) ;
    MAVAIL(plane,base)$NUMPLANE(plane,base,time) =
        MAVAIL(plane,base) + AVAIL.M(plane,base,time) );
DISPLAY PAVAIL;
DISPLAY AAVAIL;
DISPLAY MAVAIL;

************************************************************
* Compute max cargo throughput at each base onload and     *
* offload airfield (percent of capacity) and the marginal  *
* value for the maxed out airfields...                     *
************************************************************

PARAMETER PCTHRU(base) Max percent of cargo thruput used;
    PCTHRU(base) = 0;
PARAMETER MCTHRU(base) Marginal Value of Base (Cargo);
    MCTHRU(base) = 0;
LOOP(time,
    PCTHRU(base)$THRU(base,time,"blk") =
        MAX( PCTHRU(base), (-THRUPUT1.L(base,time) /
            THRU(base,time,"blk")) );
    MCTHRU(base)$THRU(base,time,"blk") =
        MCTHRU(base) + THRUPUT1.M(base,time) );
DISPLAY PCTHRU;
DISPLAY MCTHRU;
```

```
************************************************************
* Compute max passenger throughput at each base onload      *
* and offload airfield and marginal value for the maxed     *
* airfields...                                              *
************************************************************

PARAMETER PPTHRU(base) Max percent of PAX thruput used;
    PPTHRU(base) = 0;
PARAMETER MPTHRU(base) Marginal Value of Base (PAX);
    MPTHRU(base) = 0;
LOOP(time,
    PPTHRU(base)$THRU(base,time,"pax") =
        MAX( PPTHRU(base), (-THRUPUT2.L(base,time) /
            THRU(base,time,"pax")) );
    MPTHRU(base)$THRU(base,time,"pax") =
        MPTHRU(base) + THRUPUT2.M(base,time) );
DISPLAY PPTHRU;
DISPLAY MPTHRU;

************************************************************
* Compute max MOG used at each base and the marginal        *
* value for the airfields where ramp space is used up...    *
************************************************************

PARAMETER PMOG(base) Max percent of MOG used;
    PMOG(base) = 0;
PARAMETER MMOG(base) Marginal Value of Base (MOG);
    MMOG(base) = 0;
LOOP(time,
    PMOG(base)$AMOG(base,time) =
        MAX( PMOG(base), (-MOGLIM.L(base,time) / AMOG(base,time)) );
    MMOG(base)$AMOG(base,time) =
        MMOG(base) + MOGLIM.M(base,time) );
DISPLAY PMOG;
DISPLAY MMOG;

************************************************************
* Compute total amount of cargo and pax scheduled for       *
* delivery...                                               *
************************************************************

PARAMETER TOTAMT(class);
  TOTAMT(class) = SUM(unit, AMOUNT(unit,class));
DISPLAY TOTAMT;

************************************************************
* Display undelivered cargo by unit and class...            *
************************************************************

DISPLAY Z.L;
```

```
************************************************************
* Compute total amount of cargo and pax left undelivered...  *
************************************************************

PARAMETER UTOTAL(class);
  UTOTAL(class) = SUM(unit, Z.L(unit,class));
DISPLAY UTOTAL;

************************************************************
* Compute amount of cargo and pax delivered...               *
************************************************************

PARAMETER DTOTAL(class);
  DTOTAL(class) = TOTAMT(class) - UTOTAL(class);
DISPLAY DTOTAL;

************************************************************
* Compute sortie schedule by plane and day...                *
************************************************************

PARAMETER SORTIES(time,plane) ;
    SORTIES(time,plane) = SUM( (unit,route)$DELTA(unit,route,plane,time),
    X.L(time,unit,plane,"blk",route)$PREFER(plane,"blk",route) +
    X.L(time,unit,plane,"out",route)$PREFER(plane,"out",route) +
    X.L(time,unit,plane,"pax",route)$PREFER(plane,"pax",route) ) ;
DISPLAY SORTIES;

************************************************************
* Compute total number of sorties by plane type...           *
************************************************************

PARAMETER TOTALS(plane) ;
    TOTALS(plane) = SUM(time, SORTIES(time,plane));
DISPLAY TOTALS;

************************************************************
* Compute total number of sorties...                         *
************************************************************

PARAMETER TSORT;
    TSORT = SUM(plane, TOTALS(plane));
DISPLAY TSORT;

************************************************************
* Compute percent sorties flown by each plane type...        *
************************************************************

PARAMETER MIXTURE(plane);
    MIXTURE(plane) = TOTALS(plane)/TSORT;
DISPLAY MIXTURE;
```

```
************************************************************
* Compute percent of each cargo class carried by each      *
* plane type...                                            *
************************************************************

PARAMETER PERCENT(class,plane);
    PERCENT(class,plane)$DTOTAL(class) =
        SUM((unit,route,time)$(AMOUNT(unit,class)$DELTA(unit,route,plane,time)),
        PREFER(plane,class,route) * X.L(time,unit,plane,class,route)
        + NONPREF(plane,class,route) * Y.L(time,unit,plane,class,route)) / DTOTAL(class);
DISPLAY PERCENT,

************************************************************
* Compute percent of each cargo only (bulk + outsize) carried by each  *
* plane type...                                            *
************************************************************

PARAMETER PCARGO(plane);
    PCARGO(plane)$(DTOTAL("blk")+DTOTAL("out")) =
        SUM((time,unit,route,class)$(DELTA(unit,route,plane,time)
        $(AMOUNT(unit,class)$cargo(class))),
        PREFER(plane,class,route) * X.L(time,unit,plane,class,route)
        + NONPREF(plane,class,route) * Y.L(time,unit,plane,class,route))
        / (DTOTAL("blk")+DTOTAL("out"));
DISPLAY PCARGO;

************************************************************
* Compute pool of flying hours available to each plane     *
* type...                                                  *
************************************************************

PARAMETER ACTHOURS(period,base,plane);
    ACTHOURS(period,base,plane) = SUM(time$UTEPRD(period,time),
    SUM((unit,class,route,vtinc)$(FLYTIME(route,vtinc)
    $(delta(unit,route,plane,time-(ORD(vtinc)-1))
    $(PREFER(plane,class,route)$HEAD(base,route)))),
    FLYTIME(route,vtinc) *
    X.L(time-(ORD(vtinc)-1),unit,plane,class,route)) ) ;
DISPLAY ACTHOURS;

************************************************************
* Compute total number of aircraft available in each UTE period...   *
************************************************************

PARAMETER TPLANES(period,base,plane);
    TPLANES(period,base,plane) = SUM(time$(UTEPRD(period,time)
    $(NUMPLANE(plane,base,time)$HOME(base,plane))),
    NUMPLANE(plane,base,time));
```

```
************************************************************
* Compute actual utilization rate of each plane type...    *
************************************************************

PARAMETER ACTUTE(period,base,plane);
    ACTUTE(period,base,plane)$HOME(base,plane) =
    (ACTHOURS(period,base,plane) * TUNITS) / TPLANES(period,base,plane);
DISPLAY ACTUTE;

************************************************************
* Compute the actual fly rate of the aircraft...           *
************************************************************

PARAMETER AVGHOURS(period,base,plane);
    AVGHOURS(period,base,plane) = ACTHOURS(period,base,plane) / UTENUM(period);
PARAMETER FLYRATE(period,base,plane);
    FLYRATE(period,base,plane)$AAVAIL(period,base,plane) =
    AVGHOURS(period,base,plane) / AAVAIL(period,base,plane);
```

Appendix D: Desert Shield Flight and Ground Times

TABLE 22

Estimated Flight Time Between Airfields
(Source: AMC/XPYR)

Origin	Destination	C-5	C-141	KC-10	B-747	B-707	DC-10	P-747
KDOV	KFOE	3.1	3.1	3.1	3.1	3.1	3.1	3.1
KFOE	KCEF	3.1	3.1	3.1	3.1	3.1	3.1	3.1
KCEF	LETO	7.5	7.5	7.5	7.5	7.5	7.5	7.5
LETO	OEDR	6.9	7.1	6.9	6.9	6.9	6.9	6.9
LETO	OEDF	6.9	7.1	6.9	6.9	6.9	6.9	6.9
LETO	OEJB	6.9	7.1	6.9	6.9	6.9	6.9	6.9
LETO	OEKK	6.9	7.1	6.9	6.9	6.9	6.9	6.9
OEDR	LEZA	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEDF	LEZA	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEJB	LEZA	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEKK	LEZA	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEDR	LETO	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEDF	LETO	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEJB	LETO	8.2	8.2	8.2	8.2	8.2	8.2	8.2
OEKK	LETO	8.2	8.2	8.2	8.2	8.2	8.2	8.2
LETO	KDOV	8.6	8.9	8.6	8.6	8.6	8.6	8.6
KCEF	EDAF	7.6	7.9	7.6	7.6	7.6	7.6	7.6
EDAF	OEDR	7.2	7.4	7.2	7.2	7.2	7.2	7.2
EDAF	OEDF	7.2	7.4	7.2	7.2	7.2	7.2	7.2
EDAF	OEJB	7.2	7.4	7.2	7.2	7.2	7.2	7.2
EDAF	OEKK	7.2	7.4	7.2	7.2	7.2	7.2	7.2
EDAR	OEDR	6.9	7.1	6.9	6.9	6.9	6.9	6.9
EDAR	OEDF	6.9	7.1	6.9	6.9	6.9	6.9	6.9
EDAR	OEJB	6.9	7.1	6.9	6.9	6.9	6.9	6.9
EDAR	OEKK	6.9	7.1	6.9	6.9	6.9	6.9	6.9
OEDR	EDAF	7.8	8.4	7.8	7.8	7.8	7.8	7.8
OEDF	EDAF	7.8	8.4	7.8	7.8	7.8	7.8	7.8
OEJB	EDAF	7.8	8.4	7.8	7.8	7.8	7.8	7.8
OEKK	EDAF	7.8	8.4	7.8	7.8	7.8	7.8	7.8
OEDR	EDAR	8.0	8.2	8.0	8.0	8.0	8.0	8.0
OEDF	EDAR	8.0	8.2	8.0	8.0	8.0	8.0	8.0
OEJB	EDAR	8.0	8.2	8.0	8.0	8.0	8.0	8.0
OEKK	EDAR	8.0	8.2	8.0	8.0	8.0	8.0	8.0
EDAF	KDOV	9.3	9.6	9.3	9.3	9.3	9.3	9.3
EXXX	KDOV	9.3	9.6	9.3	9.3	9.3	9.3	9.3
KCHS	KFOE	2.5	2.5	2.5	2.5	2.5	2.5	2.5
KFOE	KWRI	2.7	2.7	2.7	2.7	2.7	2.7	2.7
KWRI	LETO	7.4	7.7	7.4	7.4	7.4	7.4	7.4
LETO	KCHS	9.8	9.8	9.8	9.8	9.8	9.8	9.8
LEZA	KCHS	9.8	9.8	9.8	9.8	9.8	9.8	9.8
LEZA	OEKK	8.2	8.2	8.2	8.2	8.2	8.2	8.2
KWRI	EDAF	8.0	8.0	8.0	8.0	8.0	8.0	8.0

EDAF	KCHS	10.0	10.6	10.0	10.0	10.0	10.0	10.0
KDOV	LETO	7.5	7.7	7.5	7.5	7.5	7.5	7.5
KDOV	EDAF	8.0	8.1	8.0	8.0	8.0	8.0	8.0
KDOV	EXXX	8.0	8.1	8.0	8.0	8.0	8.0	8.0
KCHS	KDOV	1.4	1.5	1.4	1.4	1.4	1.4	1.4
KNON	KFOE	2.5	2.5	2.5	2.5	2.5	2.5	2.5
KNON	KDOV	0.0	0.0	0.2	1.2	1.2	0.2	0.2
EXXX	EDAF	0.0	0.0	0.2	1.2	1.2	0.2	0.2
KFOE	EXXX	8.2	8.2	8.2	8.2	8.2	8.2	8.2
EXXX	OEDR	7.3	7.3	7.3	7.3	7.3	7.3	7.3
EXXX	OEDF	7.3	7.3	7.3	7.3	7.3	7.3	7.3
EXXX	OEKK	7.3	7.3	7.3	7.3	7.3	7.3	7.3
EXXX	OEJB	7.3	7.3	7.3	7.3	7.3	7.3	7.3
OEDR	EXXX	8.3	8.3	8.3	8.3	8.3	8.3	8.3
OEDF	EXXX	8.3	8.3	8.3	8.3	8.3	8.3	8.3
OEJB	EXXX	8.3	8.3	8.3	8.3	8.3	8.3	8.3
OEKK	EXXX	8.3	8.3	8.3	8.3	8.3	8.3	8.3
EXXX	KNON	9.5	9.5	9.5	9.5	9.5	9.5	9.5
EDAF	EDAF	0.2	0.2	0.2	0.2	0.2	0.2	0.2
EDAR	EDAF	0.2	0.2	0.2	0.2	0.2	0.2	0.2
EDAR	EDAR	0.2	0.2	0.2	0.2	0.2	0.2	0.2

TABLE 23

Average Ground Times by Aircraft Type
(Source: AMC/XPYR)

Aircraft Type	On-load	Enroute	Off-load
C-5	4.25	3.25	3.25
C-141	3.25	2.25	2.25
B-747	4.00	2.25	4.00
B-707	3.25	2.25	2.25
DC-10	4.00	2.25	4.00
P-747	4.00	2.25	4.00

```
******************************************************************
* Filename:     desert.gms                                       *
*                                                                *
* Purpose:      Validation version of Airlift Capabilities       *
*               Estimation Prototype (ACEP) model using data from *
*               Operation Desert Shield.                          *
*                                                                *
* Version:      1.5                        Date: 30 Jan 1993      *
******************************************************************
$OFFTEXT
$OFFSYMXREF
$OFFSYMLIST
OPTIONS ITERLIM=75000,RESLIM=25000 ;
OPTIONS LIMCOL=0,LIMROW=0 ;
OPTION SOLPRINT=OFF ;

SET day       Periods In The Model
 /C01 * C34 / ;

 alias(day,day2) ;

SET unit      Unit Requiring Movement Of Cargo Or Pax
 /U01 * U13 / ;

SET plane     Types Of Planes
 / C005, C141, B747, B707, DC10, P747 /

SET class     Types Of Cargo Or Pax
 / blk, pax, out / ;

SET cargo(class) Cargo types only
 / blk, out / ;

SET base      All Bases Considered In The Model
 / KDOV, KCEF, KCHS, KWRI, KFOE, EDAF, LETO, OEDR,
   KNON, EXXX, EDAR, LEZA, OEJB, OEKK, OEDF / ;

SET route     Routes By Which Cargo Can Move
 / R001 * R032 /;

SET vtinc     Time Increments In Paths
 /0 * 4 /;

SET delta(unit,route,plane,day) Feasible Pickup Times
 / U01.R001.C005.(C01*C30)
   U01.(R004,R007).C141.(C01*C30)
   U01.R032.P747.(C01*C30)
   U02.R002.C005.(C01*C30)
   U02.(R005,R008).C141.(C01*C30)
   U03.R003.C005.(C01*C30)
   U03.R027.P747.(C01*C30)
   U04.(R006,R009).C141.(C01*C30)
   U05.(R010,R028).(C005,B747).(C01*C30)
   U05.(R013,R016).C141.(C01*C30)
   U06.(R011,R029).(C005,B707,B747).(C01*C30)
   U06.(R014,R017).C141.(C01*C30)
   U07.(R012,R019,R030).(C005,B707,B747).(C01*C30)
   U07.(R015,R018).C141.(C01*C30)
   U08.R020.C005.(C01*C30)
   U08.R026.C141.(C01*C30)
   U08.R031.DC10.(C01*C30)
   U09.R021.C005.(C01*C30)
   U10.R022.C005.(C01*C30)
   U11.R023.C141.(C01*C30)
   U12.R024.C141.(C01*C30)
   U13.R025.C141.(C01*C30) / ;

TABLE AMOUNT(unit,class) Amount To Be Moved
           blk    pax    out
   U01    6500   6850   5670
   U02    2800   3536   3276
   U03          25930   2646
   U04    3375
   U05   36378  25160
   U06    4582   2176
   U07    9704   5168
   U08    2079   7178   1764
   U09     252   1224    882
   U10     567   1564    882
   U11    3393
   U12    1034
   U13    1307                ;

PARAMETER TCLASS(class);
 TCLASS(class) = SUM(unit, AMOUNT(unit,class));
DISPLAY TCLASS;

SCALAR RATIO / 0.86 /;

PARAMETER FLYTIME(route,vtinc) Flight Time Between Stops
 / R001.0 13.7, R001.1 17.2, R001.2 6.6
   R002.0 13.7, R002.1 17.2, R002.2 6.6
   R003.0 13.7, R003.1 17.2, R003.2 6.6
   R004.0 15.7, R004.1 20.2, R004.2 1.6
   R005.0 15.7, R005.1 20.2, R005.2 1.6
   R006.0 15.7, R006.1 20.2, R006.2 1.6
   R007.0 15.7, R007.1 20.2, R007.2 1.6
   R008.0 15.7, R008.1 20.2, R008.2 1.6
   R009.0 15.7, R009.1 20.2, R009.2 1.6
   R010.0 21.6, R010.1  9.6
   R011.0 21.6, R011.1  9.6
   R012.0 21.6, R012.1  9.6
   R013.0 16.3, R013.1 18.0
   R014.0 16.3, R014.1 18.0
   R015.0 16.3, R015.1 18.0
   R016.0 16.3, R016.1 18.0
   R017.0 16.3, R017.1 18.0
   R018.0 16.3, R018.1 18.0
   R019.0 15.3, R019.1 17.8
   R020.0 15.3
   R021.0 15.3
   R022.0 15.3
   R023.0 15.3
   R024.0 15.3
   R025.0 15.3
   R026.0 15.3
   R027.0 16.7, R027.1 17.6, R027.2 1.5
   R028.0 16.3, R028.1 17.3, R028.2 0.5
   R029.0 16.3, R029.1 17.3, R029.2 0.5
   R030.0 16.3, R030.1 17.3, R030.2 0.5
   R031.0 15.7
   R032.0 17.7, R032.1 17.8, R032.2 0.3  / ;

SET legtime(route,base,vtinc) Time Increment at Intermediate Stops
 / R001.(KDOV.0,KFOE.0,KCEF.0,LETO.1,OEDR.1,LETO.2,KDOV.2)
   R002.(KDOV.0,KFOE.0,KCEF.0,LETO.1,OEJB.1,LETO.2,KDOV.2)
   R003.(KDOV.0,KFOE.0,KCEF.0,LETO.1,OEDF.1,LETO.2,KDOV.2)
   R004.(KCHS.0,KFOE.0,KWRI.0,LETO.0,OEDR.1,LEZA.2,KCHS.2)
   R005.(KCHS.0,KFOE.0,KWRI.0,LETO.0,OEJB.1,LEZA.2,KCHS.2)
   R006.(KCHS.0,KFOE.0,KWRI.0,LETO.0,OEKK.1,LEZA.2,KCHS.2)
   R007.(KCHS.0,KFOE.0,KWRI.0,LETO.0,OEDR.1,LETO.2,KCHS.2)
   R008.(KCHS.0,KFOE.0,KWRI.0,LETO.0,OEJB.1,LETO.2,KCHS.2)
   R009.(KCHS.0,KFOE.0,KWRI.0,LETO.0,OEKK.1,LETO.2,KCHS.2)
   R010.(KDOV.0,LETO.0,OEDR.0,LETO.1,KDOV.1)
   R011.(KDOV.0,LETO.0,OEJB.0,LETO.1,KDOV.1)
   R012.(KDOV.0,LETO.0,OEKK.0,LETO.1,KDOV.1)
   R013.(KCHS.0,KDOV.0,LETO.0,OEDR.1,LEZA.1,KCHS.1)
   R014.(KCHS.0,KDOV.0,LETO.0,OEJB.1,LEZA.1,KCHS.1)
   R015.(KCHS.0,KDOV.0,LETO.0,OEKK.1,LEZA.1,KCHS.1)
   R016.(KCHS.0,KDOV.0,LETO.0,OEDR.1,LETO.1,KCHS.1)
   R017.(KCHS.0,KDOV.0,LETO.0,OEJB.1,LETO.1,KCHS.1)
   R018.(KCHS.0,KDOV.0,LETO.0,OEKK.1,LETO.1,KCHS.1)
   R019.(KNON.0,KDOV.0,EXXX.0,OEKK.1,EXXX.1,KNON.1)
   R020.(EDAF.0,OEDR.0,EDAF.1)
   R021.(EDAF.0,OEJB.0,EDAF.1)
   R022.(EDAF.0,OEKK.0,EDAF.1)
   R023.(EDAR.0,OEDR.0,EDAR.1)
   R024.(EDAR.0,OEJB.0,EDAR.1)
   R025.(EDAR.0,OEKK.0,EDAR.1)
   R026.(EDAR.0,EDAF.0,OEDR.0,EDAR.1)
   R027.(KNON.0,KFOE.0,EXXX.0,OEDF.1,EXXX.1,KNON.2)
   R028.(KNON.0,KDOV.0,EXXX.0,OEDR.1,EXXX.1,KNON.2)
   R029.(KNON.0,KDOV.0,EXXX.0,OEJB.1,EXXX.1,KNON.2)
   R030.(KNON.0,KDOV.0,EXXX.0,OEKK.1,EXXX.1,KNON.2)
   R031.(EXXX.0,EDAF.0,OEDR.0,EXXX.1)
   R032.(KNON.0,KFOE.0,EXXX.0,OEDR.1,EXXX.1,KNON.2)   /;

SET onload(unit,base) Onload Point For Each Unit
 / (U01,U02,U03,U04).KFOE
   (U05,U06,U07).KDOV
   (U08,U09,U10).EDAF
   (U11,U12,U13).EDAR /;

SET offload(unit,base) Offload Point For Each Unit
 / (U01,U05,U08,U11).OEDR
   (U02,U06,U09,U12).OEJB
   (U03).OEDF
   (U04,U07,U10,U13).OEKK / ;

PARAMETER numplane(plane,day) Number Of Aircraft Available
 / C005.(C01*C34)   112
   C141.(C01*C34)   230
   B747.(C01*C34)    10
   B707.(C01*C34)    15
   DC10.(C01*C34)     5
   P747.(C01*C34)    10   /;

SET period UTE Periods Tracked by Model
 / Sustain / ;

SET UTEdays(period,day) Days Covered by UTE Periods
 / Sustain.(C01*C30) / ;

PARAMETER UTEnum(period) Number of days in each period
 / Sustain  30 /;

TABLE UTErate(period,plane) Objective UTE Rates for Aircraft
            C141    C005   B747    B707    DC10    P747
  Sustain   10.0     9.0   10.0    10.0    10.0    10.0   ;

PARAMETER UTEhours(period,plane) ;
  UTEhours(period,plane) = SUM(day$(UTEdays(period,day)
    $NUMPLANE(plane,day)), NUMPLANE(plane,day) *
    UTErate(period,plane) ) ;

TABLE CYCLE(route,plane) Length Of Roundtrip in Days
            C141   C005   B747   B707   DC10   P747
  R001        4      4      4      4      4      4
  R002        4      4      4      4      4      4
  R003        4      4      4      4      4      4
  R004        4      4      4      4      4      4
```

```
      R005      4     4     4     4     4     4
      R006      4     4     4     4     4     4
      R007      4     4     4     4     4     4
      R008      4     4     4     4     4     4
      R009      4     4     4     4     4     4
      R010      3     3     3     3     3     3
      R011      3     3     3     3     3     3
      R012      3     3     3     3     3     3
      R013      3     3     3     3     3     3
      R014      3     3     3     3     3     3
      R015      3     3     3     3     3     3
      R016      3     3     3     3     3     3
      R017      3     3     3     3     3     3
      R018      3     3     3     3     3     3
      R019      3     3     3     3     3     3
      R020      3     3     3     3     3     3
      R021      3     3     3     3     3     3
      R022      3     3     3     3     3     3
      R023      3     3     3     3     3     3
      R024      3     3     3     3     3     3
      R025      3     3     3     3     3     3
      R026      3     3     3     3     3     3
      R027      4     4     4     4     4     4
      R028      4     4     4     4     4     4
      R029      4     4     4     4     4     4
      R030      4     4     4     4     4     4
      R031      3     3     3     3     3     3
      R032      4     4     4     4     4     4  ;

PARAMETER PARK(base,day)  Parking MOG Limits At Each Base
 / KDOV.(C01*C34)  500
   KCEF.(C01*C34)  500
   KCHS.(C01*C34)  500
   KWRI.(C01*C34)  500
   KFOE.(C01*C34)  500
   KNON.(C01*C34)  500
   EDAF.(C01*C34)  144
   EXXX.(C01*C34)  500
   EDAR.(C01*C34)   42
   LETO.(C01*C34)  160
   LEZA.(C01*C34)   24
   OEDR.(C01*C34)   40
   OEJB.(C01*C34)   15
   OEKK.(C01*C34)   31
   OEDF.(C01*C34)   40  /;

PARAMETER THRU(base,day,class)  Throughput Limits
 / KFOE.(C01*C34).pax  15000
   KFOE.(C01*C34).blk   5000
   KDOV.(C01*C34).pax  15000
   KDOV.(C01*C34).blk   5000
   EDAF.(C01*C34).pax   15000
   EDAF.(C01*C34).blk    5000
   EDAR.(C01*C34).pax   15000
   EDAR.(C01*C34).blk    5000
   OEDR.(C01*C34).pax   11000
   OEDR.(C01*C34).blk    2500
   OEJB.(C01*C34).pax    7300
   OEJB.(C01*C34).blk    1500
   OEDF.(C01*C34).pax    4100
   OEDF.(C01*C34).blk    1400
   OEKK.(C01*C34).pax    1600
   OEKK.(C01*C34).blk     450  /;

TABLE MOG(base,plane)  Mog Used By Each Plane Type At Each Base
           C141   C005   B747   B707   DC10   P747
   KDOV      1      2     2.1     1      2     2.1
   KCEF      1      2     2.1     1      2     2.1
   KCHS      1      2     2.1     1      2     2.1
   KWRI      1      2     2.1     1      2     2.1
   KFOE      1      2     2.1     1      2     2.1
   EDAF      1      2     2.1     1      2     2.1
   LETO      1      2     2.1     1      2     2.1
   OEDR      1      2     2.1     1      2     2.1
   KNON      1      2     2.1     1      2     2.1
   EXXX      1      2     2.1     1      2     2.1
   EDAR      1      2     2.1     1      2     2.1
   LEZA      1      2     2.1     1      2     2.1
   OEJB      1      2     2.1     1      2     2.1
   OEKK      1      2     2.1     1      2     2.1
   OEDF      1      2     2.1     1      2     2.1    ;

TABLE RHO(unit,class)  Penalty For Shortfall
          blk    pax    out
   U01    10     10     10
   U02    10     10     10
   U03    10     10     10
   U04    10     10     10
   U05    10     10     10
   U06    10     10     10
   U07    10     10     10
   U08    10     10     10
   U09    10     10     10
   U10    10     10     10
   U11    10     10     10
   U12    10     10     10
   U13    10     10     10  ;

TABLE NU(unit,plane)  Penalty For Each Sortie Flown
           C141   C005   B747   B707   DC10   P747
   U01       1      1      1      1      1      1
   U02       1      1      1      1      1      1
```

```
U03        1    1    1    1    1    1
U04        1    1    1    1    1    1
U05        1    1    1    1    1    1
U06        1    1    1    1    1    1
U07        1    1    1    1    1    1
U08        1    1    1    1    1    1
U09        1    1    1    1    1    1
U10        1    1    1    1    1    1
U11        1    1    1    1    1    1
U12        1    1    1    1    1    1
U13        1    1    1    1    1    1  ;

PARAMETER PREFER(plane,class,route) Plane Capacities For Preferred Cargo
 /C141.blk.(R001*R030)     27.5
  C141.pax.(R001*R030)    136.0
  C005.out.(R001*R030)     68.9
  C005.blk.(R001*R030)     68.9
  B747.blk.(R001*R030)     87.3
  B707.blk.(R001*R030)     41.1
  DC10.pax.(R001*R030)    235.0
  P747.pax.(R001*R030)    365.0  /;

PARAMETER NONPREF(plane,class,route) Capacities For Nonpreferred Cargo
 /C005.pax.(R001*R030)     75.0
  C141.pax.(R001*R030)     10.0  /;

FREE VARIABLES
COBJ                              Cost of the operation
DOBJ                              Total tonnage & pax delivered
;
POSITIVE VARIABLES
X(day,unit,plane,class,route)     Number Of Sorties
Y(day,unit,plane,class,route)     Opportune Movement Of Nonpreferred Cargo
Z(unit,class)                     Amount Of Cargo Left Undelivered
;

EQUATIONS
COST                              The Cost Objective Function
DELIVER                           The Flow Objective Function
DELIVPAX(unit)                    Account For All Pax Requirements
DELIVBLK(unit)                    Account For All Bulk Requirements
DELIVOUT(unit)                    Account For All Outsize Requirements
NONPREF1(route,plane,day)         Opportune Cargo On Pax Planes
NONPREF2(route,plane,day)         Opportune Pax On Cargo Planes
AVAIL(plane,day)                  Number Of Planes Available
THRUPUT1(base,day)                Cargo Thruput At Onloads And Offloads
THRUPUT2(base,day)                Pax Thruput At Onloads And Offloads
MOGLIM(base,day)                  Enroute Constraints
UTELIM(period,plane)              Objective UTE constraint on aircraft
DELRATO                           Ratio of delivered cargo to pax
;
```

```
COST.. COBJ =E= SUM((unit,class)$AMOUNT(unit,class),
   RHO(unit,class) * Z(unit,class)) + SUM((unit,class,route,plane,day)
   $(delta(unit,route,plane,day)$PREFER(plane,class,route)),
   NU(unit,plane) * X(day,unit,plane,class,route) ) ;

DELIVER.. DOBJ =E= SUM((day,unit,plane,class,route)
   $delta(unit,route,plane,day),
   PREFER(plane,class,route) * X(day,unit,plane,class,route) +
   NONPREF(plane,class,route) * Y(day,unit,plane,class,route) ) ;

DELIVPAX(unit)$AMOUNT(unit,"pax").. AMOUNT(unit,"pax") =E= Z(unit,"pax") +
   SUM((route,plane,day)$delta(unit,route,plane,day),
   PREFER(plane,"pax",route) * X(day,unit,plane,"pax",route) +
   NONPREF(plane,"pax",route) * Y(day,unit,plane,"pax",route) ) ;

DELIVBLK(unit)$AMOUNT(unit,"blk").. AMOUNT(unit,"blk") =E= Z(unit,"blk") +
   SUM((route,plane,day)$delta(unit,route,plane,day),
   PREFER(plane,"blk",route) * X(day,unit,plane,"blk",route) +
   NONPREF(plane,"blk",route) * Y(day,unit,plane,"blk",route) ) ;

DELIVOUT(unit)$AMOUNT(unit,"out").. AMOUNT(unit,"out") =E= Z(unit,"out") +
   SUM((route,plane,day)$delta(unit,route,plane,day),
   PREFER(plane,"out",route) * X(day,unit,plane,"out",route) ) ;

NONPREF1(route,plane,day)$NONPREF(plane,"blk",route)..
   SUM(unit$delta(unit,route,plane,day),
   X(day,unit,plane,"pax",route)$PREFER(plane,"pax",route) -
   Y(day,unit,plane,"blk",route)$NONPREF(plane,"blk",route) ) =G= 0.0 ;

NONPREF2(route,plane,day)$NONPREF(plane,"pax",route)..
   SUM(unit$delta(unit,route,plane,day),
   X(day,unit,plane,"out",route)$PREFER(plane,"out",route) +
   X(day,unit,plane,"blk",route)$PREFER(plane,"blk",route) -
   Y(day,unit,plane,"pax",route)$NONPREF(plane,"pax",route) ) =G= 0.0;

AVAIL(plane,day).. NUMPLANE(plane,day) =G=
   SUM((route,day2)$((ORD(day2) le ORD(day)) and
   (ORD(day2) gt (ORD(day) - CYCLE(route,plane))) ),
   SUM((unit,class) $(delta(unit,route,plane,day2)
   $PREFER(plane,class,route)), X(day2,unit,plane,class,route) ) ) ;

THRUPUT1(base,day)$THRU(base,day,"blk").. THRU(base,day,"blk") =G=
   SUM(unit$onload(unit,base),
   SUM((route,plane)$delta(unit,route,plane,day),
   PREFER(plane,"blk",route) * X(day,unit,plane,"blk",route) +
   PREFER(plane,"out",route) * X(day,unit,plane,"out",route) +
   NONPREF(plane,"blk",route) * Y(day,unit,plane,"blk",route) ) ) +

   SUM(unit$offload(unit,base),
   SUM((route,plane,vtinc) $(legtime(route,base,vtinc)
   $delta(unit,route,plane,day-(ORD(vtinc)-1))),
   PREFER(plane,"blk",route) * X(day-(ORD(vtinc)-1),unit,plane,"blk",route) +
   PREFER(plane,"out",route) * X(day-(ORD(vtinc)-1),unit,plane,"out",route) +
   NONPREF(plane,"blk",route) * Y(day-(ORD(vtinc)-1),unit,plane,"blk",route)));

THRUPUT2(base,day)$THRU(base,day,"pax").. THRU(base,day,"pax") =G=
  SUM(unit$onload(unit,base),
  SUM((route,plane)$delta(unit,route,plane,day),
  PREFER(plane,"pax",route) * X(day,unit,plane,"pax",route) +
  NONPREF(plane,"pax",route) * Y(day,unit,plane,"pax",route) ) ) +

  SUM(unit$offload(unit,base),
  SUM((route,plane,vtinc) $(legtime(route,base,vtinc)
  $delta(unit,route,plane,day-(ORD(vtinc)-1))),
  PREFER(plane,"pax",route) * X(day-(ORD(vtinc)-1),unit,plane,"pax",route) +
  NONPREF(plane,"pax",route) * Y(day-(ORD(vtinc)-1),unit,plane,"pax",route)));

MOGLIM(base,day)$PARK(base,day)..  PARK(base,day) =G=
  SUM((unit,class,route,plane,vtinc)$(legtime(route,base,vtinc)
  $(delta(unit,route,plane,day-(ORD(vtinc)-1)) $PREFER(plane,class,route))),
  MOG(base,plane) * X(day-(ORD(vtinc)-1),unit,plane,class,route) );

UTELIM(period,plane).. UTEhours(period,plane) =G=
  SUM(day$UTEdays(period,day), SUM((unit,class,route,vtinc)
  $(FLYTIME(route,vtinc)$(delta(unit,route,plane,day-(ORD(vtinc)-1))
  $PREFER(plane,class,route))), FLYTIME(route,vtinc) *
  X(day-(ORD(vtinc)-1),unit,plane,class,route)) );

DELRATO.. SUM((day,unit,plane,class,route)
  $(delta(unit,route,plane,day)$(amount(unit,class)$cargo(class))),
  PREFER(plane,class,route) * X(day,unit,plane,class,route) )
  - RATIO * SUM((day,unit,plane,route)$delta(unit,route,plane,day),
  PREFER(plane,"pax",route) * X(day,unit,plane,"pax",route)
  + NONPREF(plane,"pax",route) * Y(day,unit,plane,"pax",route) ) =G= 0.0 ;

MODEL ACEPCN / COST, DELIVPAX, DELIVBLK, DELIVOUT,
           NONPREF1, NONPREF2, AVAIL, THRUPUT1,
           THRUPUT2, MOGLIM, UTELIM, DELRATO / ;

**********************************************************
* Solve model ACEPCN to minimize cost...                 *
**********************************************************

SOLVE ACEPCN USING LP MINIMIZING COBJ ;

PARAMETER PAVAIL(plane) Max Percent of Available ACFT Used;
  PAVAIL(plane) = 0;
PARAMETER AAVAIL(period,plane) Average number used;
  AAVAIL(period,plane) = SUM(day$UTEDAYS(period,day),
  -AVAIL.L(plane,day)) / UTEnum(period) ;
PARAMETER MAVAIL(plane) Marginal Value of Aircraft;
```

```
  MAVAIL(plane) = 0;
 LOOP(day,
    PAVAIL(plane)$NUMPLANE(plane,day) =
     MAX( PAVAIL(plane), (-AVAIL.L(plane,day) /
         NUMPLANE(plane,day)) );
    MAVAIL(plane)$NUMPLANE(plane,day) =
     MAVAIL(plane) + AVAIL.M(plane,day) );
 DISPLAY PAVAIL;
 DISPLAY AAVAIL;
 DISPLAY MAVAIL;

 PARAMETER PCTHRU(base) Max percent of cargo thruput used;
  PCTHRU(base) = 0;
 PARAMETER MCTHRU(base) Marginal Value of Base (Cargo);
  MCTHRU(base) = 0;
 LOOP(day,
    PCTHRU(base)$THRU(base,day,"blk") =
     MAX( PCTHRU(base), (-THRUPUT1.L(base,day) /
         THRU(base,day,"blk")) );
    MCTHRU(base)$THRU(base,day,"blk") =
     MCTHRU(base) + THRUPUT1.M(base,day) );
 DISPLAY PCTHRU;
 DISPLAY MCTHRU;

 PARAMETER PPTHRU(base) Max percent of PAX thruput used;
  PPTHRU(base) = 0;
 PARAMETER MPTHRU(base) Marginal Value of Base (PAX);
  MPTHRU(base) = 0;
 LOOP(day,
    PPTHRU(base)$THRU(base,day,"pax") =
     MAX( PPTHRU(base), (-THRUPUT2.L(base,day) /
         THRU(base,day,"pax")) );
    MPTHRU(base)$THRU(base,day,"pax") =
     MPTHRU(base) + THRUPUT2.M(base,day) );
 DISPLAY PPTHRU;
 DISPLAY MPTHRU;

 PARAMETER PMOG(base) Max percent of MOG used;
  PMOG(base) = 0;
 PARAMETER MMOG(base) Marginal Value of Base (MOG);
  MMOG(base) = 0;
 LOOP(day,
    PMOG(base)$PARK(base,day) =
     MAX( PMOG(base), (-MOGLIM.L(base,day) / PARK(base,day)) );
    MMOG(base)$PARK(base,day) =
     MMOG(base) + MOGLIM.M(base,day) );
 DISPLAY PMOG;
 DISPLAY MMOG;

 PARAMETER TOTAMT(class);
  TOTAMT(class) = SUM(unit, AMOUNT(unit,class));
```

```
DISPLAY TOTAMT;

DISPLAY Z.L;

PARAMETER UTOTAL(class);
  UTOTAL(class) = SUM(unit, Z.L(unit,class));
DISPLAY UTOTAL;

PARAMETER DTOTAL(class);
  DTOTAL(class) = TOTAMT(class) - UTOTAL(class);
DISPLAY DTOTAL;

PARAMETER SORTIES(day,plane) ;
  SORTIES(day,plane) = SUM( (unit,route)$DELTA(unit,route,plane,day),
    X.L(day,unit,plane,"blk",route)$PREFER(plane,"blk",route) +
    X.L(day,unit,plane,"out",route)$PREFER(plane,"out",route) +
    X.L(day,unit,plane,"pax",route)$PREFER(plane,"pax",route) ) ;
DISPLAY SORTIES;

PARAMETER TOTALS(plane) ;
  TOTALS(plane) = SUM(day, SORTIES(day,plane));
DISPLAY TOTALS;

PARAMETER TSORT;
  TSORT = SUM(plane, TOTALS(plane));
DISPLAY TSORT;

PARAMETER MIXTURE(plane);
  MIXTURE(plane) = TOTALS(plane)/TSORT;
DISPLAY MIXTURE;

PARAMETER PERCENT(class,plane);
  PERCENT(class,plane) = 0.0;
  PERCENT(class,plane)$DTOTAL(class) =
    SUM((unit,route,day)$(AMOUNT(unit,class)$DELTA(unit,route,plane,day)),
      PREFER(plane,class,route) * X.L(day,unit,plane,class,route)
      + NONPREF(plane,class,route) * Y.L(day,unit,plane,class,route))
      / DTOTAL(class);
DISPLAY PERCENT;

PARAMETER PCARGO(plane);
  PCARGO(plane)$(DTOTAL("blk")+DTOTAL("out")) =
  SUM((day,unit,route,class)$(DELTA(unit,route,plane,day)
  $(amount(unit,class)$cargo(class))),
  PREFER(plane,class,route) * X.L(day,unit,plane,class,route))
  / (DTOTAL("blk")+DTOTAL("out"));
DISPLAY PCARGO;

PARAMETER ACTHOURS(period,plane);
  ACTHOURS(period,plane) = SUM(day$UTEdays(period,day),
  SUM((unit,class,route,vtinc)$(FLYTIME(route,vtinc)
  $(delta(unit,route,plane,day-(ORD(vtinc)-1))
  *PREFER(plane,class,route))), FLYTIME(route,vtinc) *
  X.L(day-(ORD(vtinc)-1),unit,plane,class,route)) ) ;

PARAMETER TPLANES(period,plane);
  TPLANES(period,plane) = SUM(day$(UTEDAYS(period,day)
  $NUMPLANE(plane,day)), NUMPLANE(plane,day)) ;

PARAMETER ACTUTE(period,plane);
  ACTUTE(period,plane) = 0.0;
  ACTUTE(period,plane)$TPLANES(period,plane) =
  ACTHOURS(period,plane) / TPLANES(period,plane) ;
DISPLAY ACTUTE;

PARAMETER AVGHOURS(period,plane);
  AVGHOURS(period,plane) = ACTHOURS(period,plane) / UTEnum(period);
DISPLAY AVGHOURS;

PARAMETER FLYRATE(period,plane);
  FLYRATE(period,plane)$AAVAIL(period,plane) =
  AVGHOURS(period,plane) / AAVAIL(period,plane);
DISPLAY FLYRATE;
```

Appendix F: ACEP Desert Shield Solution

```
GENERAL ALGEBRAIC MODELING SYSTEM
MODEL STATISTICS    SOLVE ACEPCN USING LP FROM LINE 451

MODEL STATISTICS

BLOCKS OF EQUATIONS        12     SINGLE EQUATIONS   2020
BLOCKS OF VARIABLES         4     SINGLE VARIABLES   3028
NON ZERO ELEMENTS       36745

GENERATION TIME        =    162.920 SECONDS

EXECUTION TIME         =    163.810 SECONDS

SOLUTION REPORT     SOLVE ACEPCN USING LP FROM LINE 451

          SOLVE    SUMMARY

   MODEL   ACEPCN          OBJECTIVE  COBJ
   TYPE    LP            DIRECTION  MINIMIZE
   SOLVER  MINOS5             FROM LINE  451

**** SOLVER STATUS              1 NORMAL COMPLETION
**** MODEL STATUS               1 OPTIMAL
**** OBJECTIVE VALUE            143340.8916

 RESOURCE USAGE, LIMIT       5053.780    25000.000
 ITERATION COUNT, LIMIT      29695       75000

   MINOS  5.2 (Mar 1988)
   =====
   B. A. Murtagh, University of New South Wales
                          and
   P. E. Gill,  W. Murray,  M. A. Saunders and M. H. Wright
   Systems Optimization Laboratory, Stanford University.

 Work space needed (estimate)  --   204090 words.
 Work space available          --   244909 words.

 EXIT -- OPTIMAL SOLUTION FOUND


**** REPORT SUMMARY :     0   NONOPT
                          0   INFEASIBLE
                          0   UNBOUNDED
```

```
---- 466 PARAMETER PAVAIL          MAX PERCENT OF AVAILABLE ACFT USED

C005 1.000,   C141 0.183,   B747 1.000,   P747 1.000

---- 467 PARAMETER AAVAIL          AVERAGE NUMBER USED

               C005      C141      B747      P747

SUSTAIN      98.158    17.769     9.488     7.420

---- 468 PARAMETER MAVAIL          MARGINAL VALUE OF AIRCRAFT

C005      EPS,   B747 1496.000,   P747      EPS

---- 480 PARAMETER PCTHRU          MAX PERCENT OF CARGO THRUPUT USED

KDOV 0.747,   KFOE 0.379,   EDAF 0.285,   OEDR 0.610,   EDAR 0.148
OEJB 0.344,   OEKK 1.000,   OEDF 0.901

---- 481 PARAMETER MCTHRU          MARGINAL VALUE OF BASE (CARGO)

OEKK 503.087

---- 493 PARAMETER PPTHRU          MAX PERCENT OF PAX THRUPUT USED

KDOV 0.266,   KFOE 0.281,   EDAF 0.103,   OEDR 0.232,   OEJB 0.077,
OEKK 0.306,   OEDF 0.890

---- 494 PARAMETER MPTHRU          MARGINAL VALUE OF BASE (PAX)

          ( ALL  ZERO )

---- 505 PARAMETER PMOG            MAX PERCENT OF MOG USED

KDOV 0.356,   KCEF 0.172,   KCHS 0.274,   KWRI 0.073,   KFOE 0.181
EDAF 0.459,   LETO 1.000,   OEDR 1.000,   KNON 0.072,   EXXX 0.090
EDAR 1.000,   LEZA 1.000,   OEJB 1.000,   OEKK 1.000,   OEDF 1.000

---- 506 PARAMETER MMOG            MARGINAL VALUE OF BASE (MOG)

LETO  0.119,   OEDR  15970.929,   EDAR      EPS,   LEZA      EPS
OEJB  3499.166,   OEKK      EPS,   OEDF      EPS

---- 510 PARAMETER TOTAMT          TOTAL AMOUNT TO BE MOVED

BLK 71971.000,   PAX 78786.000,   OUT 15120.000
```

```
----  512 VARIABLE  Z.L                    AMOUNT OF CARGO LEFT UNDELIVERED

         BLK

U01   4928.447
U04   1885.000
U05   4007.653
U1:   3393.000

----  516 PARAMETER UTOTAL                 TOTAL SHORTFALL

BLK 14214.100

----  520 PARAMETER DTOTAL                 TOTAL AMOUNT DELIVERED

BLK 57756.900,   PAX 78786.000,   OUT 15120.000

----  527 PARAMETER SORTIES                SCHEDULE OF AIRLIFT MISSIONS

           C005         C141        B747        P747

C01      29.762                     4.066       3.150
C02      60.511                     5.934       6.850
C03      21.726
C04      29.762                     4.066
C05      20.318                     5.934       3.150
C06      41.514                    16.364
C07      29.762                     4.066
C08      25.962       16.364        5.934
C09       9.216
C10      49.200                     4.066
C11      27.800                     5.934
C12      27.500                                10.000
C13      23.240                     8.593
C14      53.760
C15      28.252                    10.000
C16      26.887       31.238
C17       5.418       10.762                   10.000
C18      36.576                    10.000
C19      31.400
C20      36.662
C21      29.906                    10.000      10.000
C22      12.656
C23      32.347
C24      27.076       16.364       10.000
C25      30.168       25.636                   10.000
C26      16.031
C27      14.031       15.000       10.000
C28      34.031
C29      22.675       12.326                   10.000
C30      41.263        8.683       10.000
```

```
--- 531 PARAMETER TOTALS          TOTAL MISSIONS BY AIRCRAFT TYPE

C005 875.412,   C141 161.329,   B747 100.000,   P747  63.150

--- 535 PARAMETER TSORT           =    1199.892

--- 539 PARAMETER MIXTURE         SHARE OF MISSIONS BY AIRCRAFT TYPE

C005 0.730,   C141 0.134,   B747 0.083,   P747 0.053

--- 548 PARAMETER PERCENT         SHARE OF CARGO BY AIRCRAFT TYPE

        C005       C141       B747       P747

BLK     0.783      0.066      0.151
PAX     0.669      0.038                 0.293
OUT     1.000

--- 556 PARAMETER PCARGO

C005 0.828,   C141 0.053,   B747 0.120

--- 573 PARAMETER ACTUTE          ACTUAL UTE RATE ACHIEVED

              C005       C141       B747       P747

SUSTAIN       8.144      0.528      10.000     7.486

--- 582 PARAMETER FLYRATE         FLY RATE ACHIEVED

              C005       C141       B747       P747

SUSTAIN       9.327      7.344      10.345     9.655

**** FILE SUMMARY

INPUT      GOR93M:[RMCCANNE.ACEP]DESERT.GMS;44
OUTPUT     GOR93M:[RMCCANNE.ACEP]DESERT.LIS;34

EXECUTION TIME    =    16.900 SECONDS
```

Bibliography

Alink, G.A., and S. Blackstone. "Validation and Verification of Computer Models: Working Toward a Standard Procedure," *Proceedings of the 31st Annual U.S. Army Operations Research (AORS) Symposium.* Section H, 3-12. Bethesda, MD: US Army Concepts Analysis Agency, November 1992.

Babb, Maj Bruce. "Desert Shield: Experiences on the MAC Crisis Action Team," *Airlift:* 1 - 3 (Winter 1990-1991).

Bazaraa, Mokhtar S., John J. Jarvis and Hanif D. Sherali. *Linear Programming and Network Flows* (Second Edition). New York: John Wiley and Sons, 1990.

Bodin, Lawrence, Bruce Golden, Arjang Assad, and Michael Ball. "Routing and Scheduling of Vehicles and Crews: The State of the Art," *Computers and Operations Research, 10*: 62 - 212 (1983).

Bossert, Capt Phil. "Desert Shield: The Increasing Importance of Strategic Mobility," *Airlift:* 3 - 4 (Fall 1990).

Brooke, Anthony, David Kendrick ard Alexander Meeraus. *GAMS: A User's Guide*. San Francisco: The Scientific Press, 1988.

Busch, Ingrid K. and Michael R. Hilliard. *An Airlift Capabilities Estimation Model for the Military Airlift Problem.* Operations Research Group, Center for Transportation Analysis, Oak Ridge National Laboratory, 1992.

Comptroller General of the United States. *Report to the Congress: Airlift Operations of MAC During 1973 Mid-East War.* Washington DC: GPO, 1975.

Conduct of the Persion Gulf War: Final Report to Congress. Washingtion DC: United States Department of Defense, April 1992.

Cooke, Capt James C. *Optimizing Force Deployment and Force Structure for the Rapid Deployment Force.* MS Thesis, AFIT/GST/OS/84M-7. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1984 (AD-A141 148).

Desrochers, M., J. K. Lenstra and M. W. P. Savelsbergh. "A Classification Scheme for Vehicle Routing and Scheduling Problems," *European Journal of Operational Research, 46:* 322 - 332 (1990).

Desrochers, Martin, Jacques Desrosiers, and Marius Solomon. "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows," *Operations Research, 40 (2)*: 342 - 354 (1992).

Gearing, Capt Rick, and Maj Jim Hill. "UTE Rates Revisited," *Airlift:* 18 - 21 (Spring 1988).

General Electric Company, Corporate Information Systems. *Software Engineering Handbook.* New York: McGraw-Hill Book Company, 1986.

Hilliard, Michael R., Rajendra S. Solanki, Cheng Liu, Ingrid K. Busch, Glen Harrison and Ronald D. Kraemer. "Scheduling the Operation Desert Storm Airlift: An Advanced Automation Scheduling Support System," *Interfaces, 22*: 131 - 146 (1992).

Hillier, Frederick S. and Gerald J. Lieberman. *Introduction to Operations Research.* Oakland, CA: Holden-Day, Inc., 1986.

Killingsworth, Maj Paul. "Estimating and Supporting Future Airlift Forces," *Airlift:* 19 - 20 (Summer 1991).

King, Capt Bob. "Transportation and Strategic Mobility," *Airlift:* 1 - 4 (Winter 1989).

Laporte, Gilbert. "The Vehicle Routing Problem: An overview of exact and approximate algorithms," *European Journal of Operational Research, 59*: 345 - 358 (1992).

Litko, Lt Col Joseph. Chief, Force Structure Analysis Division, Air Mobility Command. Scott AFB, IL. Telephone Interview. 15 July 1992.

Meyer, Scott K. "Backing-In," *The Executive Analyst, 1*: 6-8 (March 1987).

Miller, Lt Col Charles E. *Airlift Doctrine.* Maxwell AFB, AL: Air University Press, 1988.

Mitchell, Maj Wally. Chief, Eastern Plans Division, Air Mobility Command. Scott AFB, IL. Telephone Interview. 15 October 1992.

O'Keefe, Robert M., Osman Balci, and Eric P. Smith. "Validating Expert System Performance," *IEEE Expert, 2 (4):* 81 - 90 (Winter 1988).

Powell, Stewart M. "The Airlifters: They Deliver," *Air Force Magazine:* 50 - 55 (August 1991).

Pressman, Roger S. *Software Engineering: A Practitioner's Approach* (Second Edition). New York: McGraw-Hill Book Company, 1987.

Principles of Testing Student Text. Course Number 9AI-A01, DELTAK Productiviity Through Structured Testing Series 7AI-A99. U.S.A.: Applied Information Development, Inc., 1985.

Pritsker, A. Alan B. *Introduction to Simulation and SLAM II* (Third Edition). New York: John Wiley and Sons, 1986.

Rappoport, Harold K., Laurence S. Levy, Bruce L. Golden and David S. Feshbach. "Estimating Loads of Aircraft in Planning for the Military Airlift Command," *Interfaces, 21:* 63 - 78 (1991).

Rappoport, Harold K., Laurence S. Levy, Bruce L. Golden and Katherine J. Toussaint. "A Planning Heuristic for Military Airlift," *Interfaces, 22:* 73 - 87 (1992).

Ronen, David. "Perspectives on Practical Aspects of Truck Routing and Scheduling," *European Journal of Operational Research, 35:* 137 - 145 (1988).

Smith, Lt Col Ronny C. "Crisis Action System," *Airlift:* 13 -16 (Fall 1986).

Smith, Maj Ronny C. "Station Capability," *Airlift:* 14 - 16 (Winter 1985).

Solomon, Marius M. and Jacques Desrosiers. "Time Window Constrained Routing and Scheduling Problems," *Transportation Science, 22 (1):* 1 - 13 (1988).

Specht, R.D. "The Nature of Models," in *Systems Analysis and Policy Planning: Applications in Defense*. Edited by E.S. Quade and W.I. Boucher. New York: Elsevier North-Holland, Inc., 1968.

Streater, Col Don. "Direct Delivery: Power Projection Doctrine for the Nineties," *Airlift:* 14 - 17 (1988).

Ulsamer, Edgar. "The Airlift Master Plan," *Air Force Magazine:* 58 - 65 (May 1984).

Williams, H. P. *Model Building in Mathematical Programming.* New York: John Wiley and Sons, 1985.

Vita

Captain Randy McCanne was born on 4 November 1960 in Denver, Colorado. He graduated from Pueblo East High School in Pueblo, Colorado in 1979 and attended the U.S. Air Force Academy, graduating with a Bachelor of Science (specialty: Computer Science) in June 1983. Upon graduation, he received a regular commission in the USAF and attended Undergraduate Pilot Training – Helicopter at Fort Rucker, Alabama. He graduated with honors from pilot training in April 1984, and was assigned to fly HH-3E helicopters for the 71st Aerospace Rescue and Recovery Squadron, Elmendorf AFB, Alaska. In 1988, Capt McCanne was grounded for medical reasons and cross-trained into the Communications-Computer Systems career field. He was assigned to the Operating Systems Section, Air Force Global Weather Central, Offutt AFB, Nebraska as the Officer in Charge of the Real-Time Weather Team where he led the development and maintenance of real-time weather information processing software. In July 1989 Capt McCanne became the Chief of the 11 person Operating Systems Section, and in August 1991 he entered the School of Engineering at the Air Force Institute of Technology.

Permanent Address: 1340 Horseshoe Dr.
Pueblo, CO 81001

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1993

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
THE AIRLIFT CAPABILITIES ESTIMATION PROTOTYPE:
A CASE STUDY IN MODEL VALIDATION

5. FUNDING NUMBERS

6. AUTHOR(S)
Randy McCanne, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/93M-13

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ Air Mobility Command/XPYR
402 Scott Dr. Room 132
Scott AFB IL 62225-5363

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This study investigates the application of a life cycle approach to the validation of operational models. The classic "waterfall" life cycle from software engineering is adapted for use on mathematical models by defining four stages of model development. Each stage is discussed in detail and examples of the output from each stage are presented. In addition, techniques are investigated for applying the proposed life cycle to existing models through the recovery of life cycle stages.

The methodology is applied to a linear programming model developed for planning airlift operations to demonstrate the power of the life cycle approach to validation. The results of applying each stage of the life cycle to the model are presented. As a final test, the model is used to predict the airlift capability and resource requirements for the Operation Desert Shield airlift. A comparison is made between the predictions of the model and data from the actual operation. The validated model is shown to be a better representation of the airlift planning problem. Finally, specific recommendations are made for operational use of the airlift planning model and on areas where further research is needed on both the model and the life cycle validation approach.

14. SUBJECT TERMS
Airlift; Validation; Linear Programming; Mathematical Modeling

15. NUMBER OF PAGES
127

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL